TEXT MINING IN HEALTHCARE

Bringing Structure to Electronic Health Records

Ayoub Bagheri

TEXT MINING IN HEALTHCARE: Bringing Structure to Electronic Health Record

Ayoub Bagheri

Text Mining in Healthcare

Bringing Structure to Electronic Health Records

Tekst Mining in Gezondheidszorg
Structuur Aanbrengen in Elektronische Gezondheidsdossiers
(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit Utrecht
op gezag van de
rector magnificus, prof.dr. H.R.B.M. Kummeling,
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen op

vrijdag 15 januari 2021 des middags te 4.15 uur

door

Ayoub Bagheri

geboren op 18 augustus 1985
te Bojnord

Promotoren:

Prof. dr. P.G.M. van der Heijden
Prof. dr. F.W. Asselbergs

Copromotor:

Dr. D.L. Oberski

Beoordelingscommissie:

Prof. dr. M.J.C. Eijkemans	University Medical Center Utrecht
Prof. dr. J.H. Moore	University of Pennsylvania
Prof. dr. F.E. Scheepers	University Medical Center Utrecht
Prof. dr. A.G.J. van de Schoot	Utrecht University
Prof. dr. M.R. Spruit	Leiden University

Text Mining in Healthcare: Bringing Structure to Electronic Health Records
Proefschrift Universiteit Utrecht, Utrecht. - Met samenvatting in het Nederlands.

ISBN: 978-94-6416-390-2
Print: Ridderprint | www.ridderprint.nl

Contents

Chapter 1

Introduction

Electronic health records (EHRs) are rich in data with the potential to leverage applications that provide safer care, reduce medical errors, reduce healthcare expenditure, and enable providers to improve their productivity and efficiency (Jensen, Jensen, & Brunak, 2012; Mehta & Pandit, 2018; Sutton et al., 2020; Xiao, Choi, & Sun, 2018). A major portion of this data is inside free text in the form of physicians' notes, discharge summaries, and radiology reports among other types of clinical narratives. This clinical text follows the patient through the care procedures and documents the patient's complaint and symptoms, physical exam, diagnostic tests, conclusions, treatments, and outcomes of the treatment.

Free text in the clinical domain is considered as a kind of unstructured information, which is difficult to process automatically. Despite many attempts to encode text in the form of structured data (Alex et al., 2019; Baghdadi et al., 2019; Byrd, Steinhubl, Sun, Ebadollahi, & Stewart, 2014; Duarte, Martins, Pinto, & Silva, 2018; Gong et al., 2008; Jonnalagadda, Adupa, Garg, Corona-Cox, & Shah, 2017; Kocbek et al., 2016; Mujtaba et al., 2019; Pons, Braun, Hunink, & Kors, 2016; Savova et al., 2010; Sheikhalishahi et al., 2019; Sohn et al., 2014; Taira, Soderland, & Jakobovits, 2001; Z. Wang et al., 2012; Y. Wang et al., 2018; Yao, Mao, & Luo, 2019; Yim, Yetisgen, Harris, & Kwan, 2016), free text continues to be used in EHRs. Therefore, text mining techniques can be applied to create a more structured representation of a text, making its content more accessible for data science, machine learning and statistics (Mujtaba et al., 2019; Pons et al., 2016; Sheikhalishahi et al., 2019; Y. Wang et al., 2018; Yim et al., 2016). To this end, this thesis is concentrated on providing solutions for some of the challenges in the analysis of the free text element frequently included in the clinical domain.

A widely accepted definition of text mining has been provided by Hearst (1999), as "the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources". Text mining is about looking for patterns in text, in a similar way that data mining can be

loosely described as looking for patterns in data. Clinical text mining, in the literature, is often defined more pragmatic than the main definition of text mining. Fleuren and Alkema (2015) describe clinical text mining as automated processing and analysis of text in relevant textual biomedical resources. Text mining typically involves a number of distinct phases including information retrieval, named entity recognition, information extraction and knowledge discovery. The first step concerns collecting relevant documents. After information retrieval, the resulting document collection can be analyzed by search algorithms for the occurrence of specific keywords of interest. As a last step, information extraction is performed with text data as input, resulting in structured data. Machine learning and data mining techniques are applied to the structured information to detect relations between concepts in the text.

Text mining appears to embrace the use of natural language processing (NLP) techniques. According to Mehta and Pandit (2018), NLP is one of the most widely used big data analytical techniques in healthcare, and is defined as "any computer-based algorithm that handles, augments, and transforms natural language so that it can be represented for computation" (Yim et al., 2016). There is therefore often an overlap of the tasks, methods, and goals for text mining and NLP, and the concepts are sometimes used interchangeably.

Text mining and NLP techniques have been applied to numerous health applications involving text de-identification tools (Menger, Scheepers, van Wijk, & Spruit, 2018), clinical decision support systems (Sutton et al., 2020), patient identification systems (Byrd et al., 2014; Jamian, Wheless, Crofford, & Barnado, 2019; Jonnalagadda et al., 2017; X. Wu, Zhao, Radev, & Malhotra, 2020), disease classification systems (Kocbek et al., 2016; Koopman, Karimi, et al., 2015; Torii et al., 2015), and the prediction models (K. Huang, Altosaar, & Ranganath, 2019; Jonnagaddala et al., 2015; Liu, Zhang, & Razavian, 2018; Mullenbach, Wiegreffe, Duke, Sun, & Eisenstein, 2018). Many of these clinical applications depend on some form of text classification. Text classification is a task that consists in automatically assigning a document to a predefined set of classes or labels. The focus of the work described in this thesis is the analysis and classification of the clinical free text found in heath records used in the University Medical Center (UMC) Utrecht.

Text classification in the clinical domain is more challenging than those in other general domains because of four reasons. The first reason is the lack of significant gold standard text data sets in this domain. Researchers and physicians invest intense manual effort in the creation of data sets for each task of the clinical text mining. Second, the clinical texts generally contain high levels of sparsity and noise. Sentences in the clinical domain are short and have a small number of words, with less semantic knowledge, compared to those in other domains. To address these two issues, in Chapter 2, we propose a smoothing method to enrich the short sparse text data in EHRs, before the final step of text classification.

This method is an unsupervised learning approach, and combines unlabeled clinical notes with the available small set of labeled notes. Furthermore, in Chapter 3, we extend the unsupervised method and implement a new extensive *Python* package to tackle the aforementioned issues. The software package mitigates the classification error inherent in the clinical short texts by interpolating between observed and fitted counts obtained by several clustering algorithms. Third, the clinical texts are high dimensional data and, therefore, it is difficult to integrate them with classical clinical variables in health records, for the purpose of a multimodal classification system. Recently, for this challenge, deep learning strategies have gained increased attention (W. Chen et al., 2020; Liu et al., 2018; Scheurwegs, Luyckx, Luyten, Daelemans, & Van den Bulcke, 2016; Suresh et al., 2017; Xu et al., 2018). However, in Chapter 4, we conclude that there are a very few studies that investigated multimodal data for a clinical text classification system. We then propose a deep learning method to use a dense text representation to combine with the clinical variables. Lastly, the clinical texts can be associated with more than one label. This is sometimes called extreme classification (Bengio, Dembczynski, Joachims, Kloft, & Varma, 2019), yet an open problem of machine learning. Extreme classification is a rapidly growing research area within machine learning focusing on multi-class and multi-label problems involving a large number of labels. An example of extreme classification in clinical text mining is the automated coding of international classification of diseases and related health problems (ICD). To address this issue, a detailed pre-analysis of data set should be performed to reveal the properties of labels and texts in clinical text classification task. In Chapter 5, we apply several state-of-the-art deep learning techniques to address the issue of extreme classification using 608 ICD codes as the class labels. Subsequently, in Chapter 6, we extend the proposed deep learning pipeline and limit the number of labels in the multi-label text classification task to enhance clinical usability of the final model.

The remainder of this introductory chapter provides an overview of background information that is relevant in order to understand the contents of this thesis. Section 1.1 presents the basic definitions in a text classification system. We review fundamentals of unsupervised learning, used in this thesis, in Section 1.2. In Section 3, we furthermore discuss the word embedding models. Definitions of convolutional neural networks, and recurrent neural networks are discussed in Section 1.4 and in Section 1.5, respectively. Section 1.6 presents the process of model training. We discuss hyperparameter tuning in Section 1.7. Finally, Section 1.8 gives a general overview of the organisation of this thesis.

1.1 Text Classification

Text classification is the task of assigning one or more predefined categories to text documents based on their contents. Given a text document d and a set of n_C class labels $C_L \in \{1, ..., n_C\}$, text classification tries to learn a classification function $f : d \rightarrow C_L$ that maps a document to labels. Text classification can be implemented as an automated process involving none or a small amount of interaction with expert users (Kowsari et al., 2019). A general pipeline for a text classification system is illustrated in Figure 1.1.

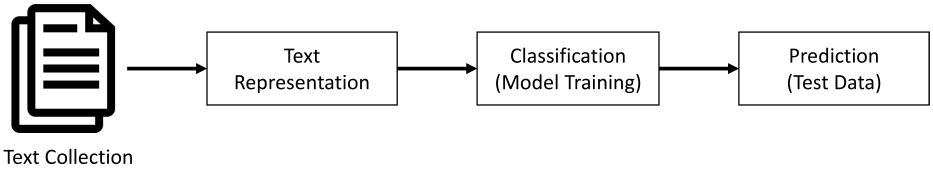


Figure 1.1: The general pipeline of a text classification system.

In single-label text classification, a document can only be related to one specific label. In binary text classification each document is assigned to either a specific predefined label or to the complement of that label. On the other hand, multi-class classification refers to the situation where each document is assigned a label from a set of n classes (where $n > 2$). Multi-label text classification refers to the case in which a document can be associated with more than one label. Text classification contains four different levels of scope that can be applied: (1) Document level, (2) Paragraph level, (3) Sentence level, and (4) Phrase level.

1.1.1 The Bag-of-Words

To perform text classification, the first question is how to represent each text document (Eisenstein, 2018; Aggarwal, 2018). A document can be seen as an observation in the data set, e.g. a patient discharge letter in a collection of discharge summaries, or a chest x-ray report. A common approach is to use vector models of a co-occurrence matrix. A co-occurrence matrix is a way of representing how often words co-occur. An example of such co-occurrence matrices is a document-term matrix, in which each row represents a document from the data set and each matrix column represents a word in the vocabulary of the data set. Table 1.1 shows a small selection from a document-term matrix of radiology reports showing the occurrence of seven words in five documents. Terms (words) used in the matrix, with their corresponding translation, are: afwijkingen (abnormalities), aortae (aortae), cag (coronary artery angiography), mogelijk (possible), nicotine (nicotine), pijn (pain), and thoracale (thoracic).

In Table 1.1, each document is represented as a vector of word counts. This

Table 1.1: Document-term matrix.

Document	afwijkingen	aortae	cag	mogelijk	nicotine	pijn	thoracale
1	1	0	1	0	0	0	0
2	1	1	1	1	0	0	0
3	1	0	0	0	1	0	1
4	1	0	0	0	0	0	0
5	1	0	0	1	0	1	1

representation is often called a bag-of-words, because it includes only information about the count of each word, and not the order in which the words appear. With the bag-of-words representation, we are ignoring grammar and order of the words. Yet the bag-of-words model is surprisingly effective for text classification (Aggarwal, 2018).

There are three commonly used bag-of-words representations of text data, corresponding to the binary model, the TF model, and the TFiDF model. A binary representation model corresponds to whether or not a word is present in the document. In some applications, such as finding frequently co-occurring groups of k words, it is sufficient to use a binary representation. However, it may lead to the loss of information because it does not contain the frequencies of the words (Eisenstein, 2018).

The most basic form of frequency-based text feature extraction is TF. TF stands for the term frequency. In this method, each word is mapped to its number of occurrences in the text. However, this approach is limited by the fact that particular words that are commonly used in the language may dominate such representations.

Most representations of text use normalized frequencies of the words. One approach is referred to as the TFiDF, where iDF stands for the inverse document frequency. The mathematical representation of the weight of the term t in the document d by TFiDF is given in:

$$TFiDF(d,t) = TF(d,t)log\left(\frac{N}{DF(t)}\right) \tag{1.1}$$

where $TF(d,t)$ is the frequency of the term t in document d, N is the number of documents and $DF(t)$ is the number of documents containing the term t. Although TFiDF tries to overcome the problem of common words in the document, it still suffers from the fact that it cannot account for the order of the words and the similarity between them in the document since each word is independently presented. Another issue with TFiDF is that even though it removes common words, it might decrease the performance by increasing the frequencies of misspellings that were not properly handled at the preprocessing step (Aggarwal, 2018; Kowsari et al.,

2019).

1.1.2 Supervised Learning

To predict a label from a bag-of-words model, we can assign a score to each word in the vocabulary, measuring the compatibility with the label. For example, for the label *healthy*, we might assign a positive score to the word "prima (fine)", and a negative score to the word "pijn (pain)". These scores are called weights (coefficients). Using annotated labels, we can automatically acquire weights using supervised machine learning. With supervised learning, the starting point is a training set D of observations, where each observation consists of an input x and an output y. Each observation x_i in the training set can be represented by a vector of features $x = [x_1, ..., x_n]$ describing the observation. And the aim for supervised learning is to learn a function f that models the relationship between the output class and the input features $y = f(x)$. We can imagine that there exists a true function f that perfectly captures the relationship between the input features x and the output class label y. But as training data is a limited sample representing reality with some degree of abstraction and simplification, the learned function will always be an approximation of the true function (Murphy, 2012).

In supervised learning, a classifier learns to approximate f by adjusting a model in such a way that it fits the training observations. For a simple linear model with positive numerical features and two classes $y = [healthy, unhealthy]$, the learning consists of finding appropriate weights w for each of the features. A large positive weight for a feature will indicate that this feature correlates with the positive (healthy) class, and a negative weight shows correlation with the negative (unhealthy) class. When the dot product of w and x exceeds some threshold t for a training observation, the observation will be considered as belonging to the positive class.

1.2 Unsupervised Learning

With unsupervised methods, there are no labeled examples to learn from, instead the goal is to find some structure or patterns in the input data (Murphy, 2012). Text clustering is an example of unsupervised learning, which aims to group texts or words according to some measure of similarity (Aggarwal, 2018).

The goal of clustering is to identify the underlying structure of the observed data, such that there are a few clusters of points, each of which is internally coherent. Clustering algorithms assign each data point to a discrete cluster $c_i \in 1, 2, ..., K$.

One of the best known clustering algorithms is k-means. Algorithm 1.1 shows the steps in the k-means algorithm (Eisenstein, 2018). After specifying the number of clusters K, the k-means algorithm assigns a cluster for each instance, and

computes a center for each cluster by averaging on instances in the cluster. K-means iterates between updates to the assignments and the centers.

Algorithm 1.1: K-means algorithm

```
Input: x_{1:N}, K
Output: z^i
for i ∈ 1, ..., N do
    z^i ← Randomly initialize cluster memberships among K clusters
end
for until converged do
    for k ∈ 1, ..., K do
        v_k ← 1/δ(z^i = k) Σ_{i=1}^{N} δ(z^i = k) x^i
    end
    for i ∈ 1, ..., N do
        z^i ← argmin_k ||x^i − v_k||^2
    end
end
```

An important property of K-means is that the converged solution depends on the initialization, and a better clustering can sometimes be found simply by re-running the algorithm from a different random starting point. Soft clustering, in contrast, instead of directly assigning each point to a specific cluster, assigns to each point a distribution over clusters (Eisenstein, 2018).

Topic modeling is particularly a similar method to soft clustering (Blei, Ng, & Jordan, 2003; Reed, 2012). Multiple clusters are associated with each document in soft clustering, similar to the topics inferred by topic modeling. Topic models can be thought of as soft clustering methods that discover probabilistic associations of documents to latent topics.

Topic modeling provides a convenient unsupervised way to analyze high-dimensional data such as text. It is a form of text analysis used to explore relationships between words within a document where the words are grouped together to form topics (clusters). In other words, a topic contains a cluster of words that frequently occurs together.

The main idea of topic models is that documents are mixtures of topics, where a topic is a probability distribution over words. Topic modeling specifies a simple probabilistic procedure by which documents can be generated. Figure 1.2 shows a text generation process by a topic model. Topic 1 and topic 2 shown in the figure have different word distributions so that they can constitute documents by choosing the words which have different importance degree to the topic. Document 1 and document 3 are generated by the respective random sampling of topic 1 and topic 2. But, topic 1 and topic 2 generate document 2 according to the mixture

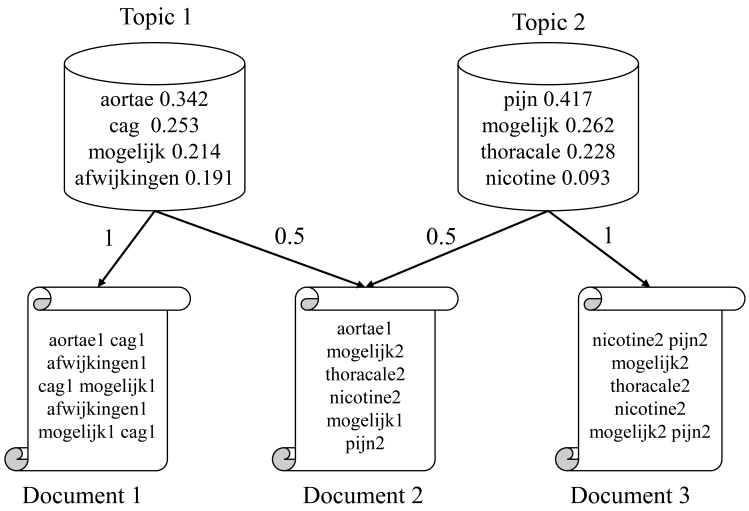


Figure 1.2: The generative process of the topic model.

of their different topic distributions (50% each). Here, the numbers at the right side of a word are its belonging topic numbers. And, the word is obtained by the random sampling of the numbered topic.

1.2.1 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a powerful generative latent topic model (Blei et al., 2003). It applies unsupervised learning on texts to induce sets of associated words. The LDA defines every topic as a distribution over the words of the vocabulary, and every document as a distribution over the topics.

LDA uses a K-dimensional latent random variable which obeys the Dirichlet distribution to represent the topic mixture ratio of the document, which simulates the generation process of the document. Let K be the multinomial topic distributions for the data set containing V elements each, where V is the number of terms in the data set. Let β_i represent the multinomial for the i-th topic, where the size of β_i is V. Given these distributions, the LDA generative process is as follows:

Algorithm 1.2: Generative process in LDA

```
for each document do
    (a) Randomly choose a K-dimensional multinomial distribution
      over topics
    for each word in the document do
        (i) Probabilistically draw β_j from the distribution over topics
          obtained in (a)
        (ii) Probabilistically draw one of the V words from β_j
    end
end
```

The LDA generative model emphasizes that documents contain multiple topics. For instance, a discharge letter might have words drawn from the topic related to the patient's symptoms and words drawn from the topic related to the patient's treatment. LDA uses sampling from the Dirichlet distribution to generate a text with the specific topic multinomial distribution, where the text is usually composed of some latent topics. And then, these topics are sampled repeatedly to generate each word for the document. Thus, the latent topics can be seen as the probability distribution of the words in the LDA model. And, each document is expressed as the random mixture of these latent topics according to the specific proportion.

The goal of LDA is to automatically discover the topics from a collection of documents. Standard statistical techniques can be used to invert the generative process of LDA, thus inferring the set of topics that were responsible for generating a collection of documents. The exact inference in LDA is generally intractable, therefore approximate inference algorithms are needed for posterior estimation. The most common approaches that are used for approximate inference are expectation-maximization, Gibbs sampling and variational method (Bagheri, Saraee, & De Jong, 2014).

1.3 Word Embedding: Distributional Hypothesis

"You shall know a word by the company it keeps."

This is a quote by Firth (1957), denoting that words occurring in similar contexts tend to have similar meanings. It outlines the idea in NLP that a statistical approach, that considers how words and phrases are used in text documents, might replicate the human notions of semantic similarity. This idea is known as the distributional hypothesis.

Word embeddings are a distributional semantics representation of words usually obtained from text data through unsupervised learning. The most common algorithm to obtain word embeddings, called word2vec, is a neural network-based

model. Word2vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Mikolov, Chen, Corrado, & Dean, 2013) includes two main algorithms: continuous bag-of-words (CBOW) and skip-gram.

1. CBOW: Predicting target word from contexts.
   This model tries to predict the tth word, w_t, in a sentence using a window of width C around the word. Therefore, the context words $w_{t-C}, w_{t-C+1}, ..., w_{t-1}, w_{t+1}, ..., w_{t+C-1}, w_{t+C}$ are at the input layer of the neural network model to predict the target word w_t.

2. Skip-gram: Predicting contexts from target word.
   This model is the opposite of the CBOW model. The target word is at the input layer, and the context words are on the output layer.

1.3.1 Continuous Bag-of-Words

The CBOW model is similar to a feed-forward neural network, where the hidden layer is removed and the projection layer is shared for all words. The model architecture is shown in Figure 1.3.

The model receives as input context words and seeks to predict the target word w_t by minimizing the CBOW loss function:

$$L_{CBOW} = -\frac{1}{|C|}\sum_{t=1}^{|C|} logP(w_t|w_{t-C}, ..., w_{t-1}, w_{t+1}, ..., w_{t+C}) \tag{1.2}$$

$P(w_t|w_{t-C}, ..., w_{t-1}, w_{t+1}, ..., w_{t+C})$ is computed using the softmax function:

$$P(w_t|w_{t-C}, ..., w_{t-1}, w_{t+1}, ..., w_{t+C}) = \frac{exp(\hat{x_t}^T x_s)}{\sum_{i=1}^{|V|} exp(\hat{x_i}^T x_s)} \tag{1.3}$$

where x_i and $\hat{x_i}$ are the word and context word embeddings of word w_i respectively. x_s is the sum of the word embeddings of the words $w_{t-C}, ..., w_{t-1}, w_{t+1}, ..., w_{t+C}$, and V is the vocabulary of the text data set.

Mikolov, Sutskever, et al. (2013) called the CBOW model a bag-of-words because the order of the context words does not influence the projection. It is also called continuous, because rather than conditioning on the words themselves, we condition on a continuous vector constructed from the word embeddings.

1.3.2 Skip-Gram

The skip-gram model is similar to CBOW, but instead of predicting a word based on the context, the context is predicted from the word. More precisely, the skip-gram architecture can be seen as a neural network without a hidden layer. It uses

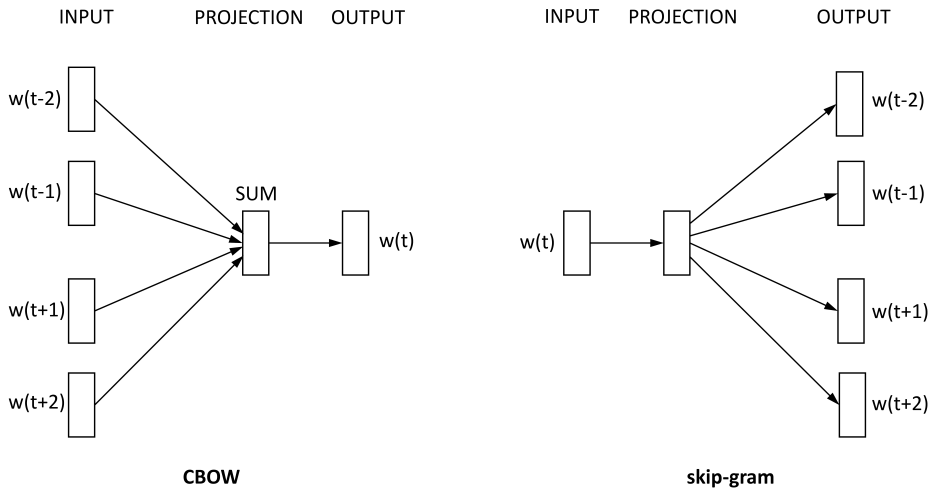


Figure 1.3: Model architectures for the CBOW and the skip-gram model (Mikolov, Sutskever, et al., 2013).

each word as input to the network to predict words within a certain range before and after that word (context size). This yields to the loss function:

$$L_{Skip-Gram} = -\frac{1}{|C|}\sum_{t=1}^{|C|}\sum_{-C\leq j\leq C, j\neq 0} logP(w_{t+j}|w_t) \tag{1.4}$$

$P(w_{t+j}|w_t)$ is computed using the softmax function:

$$P(w_{t+j}|w_t) = \frac{exp(\hat{x_{t+j}}^T x_t)}{\sum_{i=1}^{|V|} exp(\hat{x_i}^T x_t)} \tag{1.5}$$

The skip-gram architecture is shown in Figure 1.3. In this architecture, each word is generated multiple times; each time it is conditioned only on a single word. Increasing the context size in the skip-gram model increases the computational complexity, but it also improves quality of the resulting word vectors.

1.4 Convolutional Neural Networks

Convolutional neural networks (CNNs) are neural networks that use a convolution operation along with matrix multiplication operations to compute their output (Goodfellow, Bengio, & Courville, 2016). The notion of convolution filters (CF) is the central concept in CNNs that are traditionally used in signal processing (Ruder, 2019).

The main principle in a CNN model is to learn several CFs in each convolutional

layer, which are able to extract useful features from a document for the specific classification task based on the training data set. The convolutional layer slides filters of different window sizes over the concatenation of d-dimensional input word embeddings $x_1, x_2, ..., x_T$. Each filter with weights $W \in R^{kd}$ generates a new feature c_i for each window of k words:

$$c_i = \sigma(w.x_{i:i+k-1} + b) \tag{1.6}$$

where σ is an activation function (mostly the rectified linear function: ReLU) and b is the bias term for the convolution filter. Convolving the filter over the entire document yields a feature map $c \in R^{T-k+1}$:

$$c = [c_1, ..., c_{T-k+1}] \tag{1.7}$$

Each entry in the feature map is thus the result of performing a calculation that considers only a small segment of the input sequence and by applying the same filter shares parameters with the calculations to its left and right. Max-pooling, $max(c)$, is typically applied to condense a feature map to its most important feature. Max-pooling also overcomes the issue of varying document lengths (Ruder, 2019). By learning several CFs, the maximum values of the feature maps produced by them are then concatenated to a vector of filters. This vector is then fed to the next layer or an output layer.

1.5 Recurrent Neural Networks

A recurrent neural network (RNN) is a natural generalization of feed-forward neural networks to sequence data such as text. In contrast to a feed-forward neural network, however, it accepts a new input at every time step (layer). Specifically, the RNN maintains a hidden state h_t, which represents its memory of the contents of the sequence at each time step t. The RNN performs the following operation, at every time step:

$$\begin{aligned} h_t &= \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \\ y_t &= \sigma_y(W_y h_t + b_y) \end{aligned} \tag{1.8}$$

where σ is an activation function, W and U are weight matrices, h_{t-1} represents the previous hidden state, h_t shows the new hidden state, b is the bias and y_t is the output produced by the RNN at time step t.

1.5.1 Bidirectional Recurrent Neural Networks

In an RNN network, the hidden state layer captures information from the past, that is, from the beginning of the sequence up to the time step t. In contrast, in a bidirectional recurrent neural network, there is an additional layer that captures information from the future, that is, from the end of the sequence back to time step t. This architecture allows the output units to compute a representation that is dependent on both the past and the future. The hidden state h_t is the concatenation of the hidden states from the forward and backward RNNs at time step t:

$$\mathrm{h}_t = [h_{forward}; h_{backward}] \tag{1.9}$$

1.5.2 Long Short-term Memory

Long-short term memory (LSTM) (Hochreiter & Schmidhuber, 1997) networks are a variant of RNNs. The LSTM introduces mechanisms to decide what should be remembered and what should be forgotten in learning from text documents. The LSTM augments the RNN with:

- A forget gate f_t: The purpose of this gate is to delete information from the context that is no longer needed,
- An input gate i_t: To select the information to add to the current context,
- An output gate o_t: Which is used to decide what information is required for the current hidden state.

These gates are all functions of the current input x_t and the previous hidden state h_t. These gates interact with the previous cell state c_{t-1}, the current input, and the current cell state c_t and enable the model to selectively retain or overwrite information. The entire model is defined as follows:

$$\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned} \tag{1.10}$$

where σ_g is the sigmoid activation function, σ_c and σ_h are the *tanh* activation function, and $\circ$ is the element-wise multiplication.

1.5.3 Gated Recurrent Unit

The LSTM architecture is very effective, but also quite complicated. LSTM introduces a considerable number of additional parameters. Training these additional parameters makes it hard to analyze, and also imposes a much significantly higher training cost. The gated recurrent unit (GRU) was then introduced by Cho, Van Merriënboer, Bahdanau, and Bengio (2014) as an alternative to the LSTM. GRU modifies the LSTM architecture by using of a separate context vector, reducing the number of gates to two, a reset gate r_t and an update gate z_t.

$$\begin{aligned} z_t &= \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \\ r_t &= \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \\ \hat{h_t} &= \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \hat{h_t} \end{aligned} \tag{1.11}$$

where $\hat{h_t}$ is the candidate representation for the new hidden state at time t.

The purpose of the reset gate is to decide which aspects of the previous hidden state are relevant to the current context and what can be ignored. The job of the update gate is to determine which aspects of this new state will be used directly in the new hidden state and which aspects of the previous state need to be preserved for future use.

1.6 Model Training

To learn parameters of a model, we apply gradient descent using error backpropagation to minimize the loss function, and nudging the parameters in the opposite direction of the gradient. Model training proceeds by taking a text document as input, starting with random weights, and then iteratively moving through the text predicting each word w_t. The cross-entropy (negative log-likelihood) loss at each word w_t is then computed as:

$$L = -logp(w_t|w_{t-1}, ..., w_{t-n+1}) \tag{1.12}$$

The gradient for the cross-entropy loss is then:

$$\theta_{t+1} = \theta_t - \eta \frac{\partial - logp(w_t|w_{t-1}, ..., w_{t-n+1})}{\partial \theta} \tag{1.13}$$

Training the parameters to minimize loss will result in a trained model for text classification, also word embeddings that can be used as representations for other NLP tasks.

1.7 Hyperparameter Tuning

Tuning of hyperparameters is important (J. Friedman, Hastie, & Tibshirani, 2001). Hyperparameters are parameters that are chosen by the algorithm designer. In a feed-forward neural network design, hyperparameters include the learning rate, the batch size, the number of layers, the number of hidden nodes per layer, the choice of activation functions, and so on. The goal is to tune hyperparameters so that the learning classifier performs well on unseen data. For this reason, optimal values for hyperparameters are tuned on a development (tuning) set.

To predict the performance of the classifier on unseen data, also a separate subset of data (test set) must held out. It is crucial that the test set not overlap with either the training or development sets. Otherwise the model will overestimate the performance that will achieve on unlabeled data in the future. Because annotated data is expensive, this ideal can be hard to follow in practice. When only a small amount of labeled data is available, the test set accuracy can be unreliable. K-fold cross validation is one way to cope with this scenario. In this cross validation method, the labeled data is divided into K folds, and each fold acts as the test set, while training on the other folds. The test set accuracies are then aggregated. To perform hyperparameter tuning in the context of cross validation, another fold can be used for grid search. It is important not to repeatedly evaluate the cross validated accuracy while making design decisions about the classifier, or you will overstate the accuracy on truly unseen data.

1.8 Outline and Summary

The successive chapters of this thesis are organised as follows:

Chapter 2 describes the proposed method for the first research problem which pursue the extraction of medical history by clinical sentence classification. Because of the limited number of words used in clinical sentences, this problem is considered that of short text classification. To mitigate the error in short text classification, we propose an unsupervised topic modeling-based smoothing method that uses an internal knowledge acquisition mechanism without employing any external dictionary. We employ this method to enrich the representation of the sentences in a data set of clinical cardiovascular notes from the Cardiology department at the UMC Utrecht.

The considerable amount of unstructured short text in healthcare applications, particularly in clinical cardiovascular notes, has created an urgent need for tools that can parse specific information from text reports. **Chapter 3** presents a *Python*-based software package (SALTClass) for classification of short and long clinical texts. The SALTClass package is a machine learning-based NLP toolkit. It contains several functions for text preprocessing, cleaning, clustering, and classifi-

cation. The SALTClass package proposes smoothing methods based on seven clustering algorithms, namely, LDA, K-Means, MiniBatchK-Means, BIRCH, MeanShift, DBScan, and GMM. Smoothing methods are applied to the resulting cluster information to enrich the representation of sparse text. For the subsequent prediction step, ten different supervised classifiers have also been integrated into SALTClass. The SALTClass software package enables users to apply various configuration combinations to their case study. To evaluate the effectiveness of SALTClass, we analyze the cardiovascular notes collected in the UMC Utrecht.

Chapter 4 proposes a multimodal learning architecture used in a text mining pipeline to predict the recurrence of major cardiovascular events. The aim of this chapter is to demonstrate the value of clinical text classification when text data are available in addition to patients' clinical data. We propose a deep learning-based architecture that integrates neural text representation with preprocessed clinical predictors for cardiovascular risk prediction. We evaluate the added value of text data from the x-ray radiology reports to EHR data of the patients with vascular disease or a vascular risk factor, using the proposed text mining pipeline.

Chapter 5 benchmarks the state-of-the-art deep learning-based classification systems for ICD-10 coding, along with baseline systems on a data set constructed from the Dutch cardiology discharge letters at the UMC Utrecht. The ICD coding task is challenging due to the use of free text, the multi-label setting of diagnosis codes, and the large number of codes. In this chapter, we investigate models of support vector machine, CNN, LSTM, BiLSTM, and a hierarchical attention-based GRU.

Chapter 6 aims to continue the research line in Chapter 5 to create a high performing deep learning pipeline for the automated multi-label classification of reliable ICD-10 codes in the clinical free text in the cardiology domain. In this chapter, we focus on the frequently used and well defined three-digit ICD-10 codes. We investigate the usage of solely the summary paragraph of discharge letters (conclusion), adding clinical variables (age / sex) and multi-label classification as is the case in clinical practice.

Chapter 7 finally contains the discussion points where I summarize some important considerations for text mining in healthcare. In this chapter, I reflect my own personal thoughts and the work provided in this thesis on the relationship between data science and healthcare.

Chapter 2

ETM: Enrichment by Topic Modeling for Automated Clinical Sentence Classification to Detect Patients' Disease History

Bagheri, A., Sammani, A., Van der Heijden, P. G. M., Asselbergs, F. W., & Oberski, D. L. (2020). ETM: Enrichment by Topic Modeling for Automated Clinical Sentence Classification to Detect Patients' Disease History. *Journal of Intelligent Information Systems.* 55(2), 329-349. doi.org/10.1007/s10844-020-00605-w

Abstract

Given the rapid rate at which text data are being digitally gathered in the medical domain, there is growing need for automated tools that can analyze clinical notes and classify their sentences in electronic health records (EHRs). This study uses EHR texts to detect patients' disease history from clinical sentences. However, in EHRs, sentences are less topic-focused and shorter than that in general domain, which leads to the sparsity of co-occurrence patterns and the lack of semantic features. To tackle this challenge, current approaches for clinical sentence classi-

Author contributions: AB and DO designed the study. AB developed the statistical methods and source code. AB and AS analyzed the data and experiments. AB wrote the paper. AS, PvdH, FWA and DLO provided feedback on written work.

fication are dependent on external information to improve classification performance. However, this is implausible owing to a lack of universal medical dictionaries. This study proposes the ETM (enrichment by topic modeling) algorithm, based on latent Dirichlet allocation, to smoothen the semantic representations of short sentences. The ETM enriches text representation by incorporating probability distributions generated by an unsupervised algorithm into it. It considers the length of the original texts to enhance representation by using an internal knowledge acquisition procedure. When it comes to clinical predictive modeling, interpretability improves the acceptance of the model. Thus, for clinical sentence classification, the ETM approach employs an initial TFiDF (term frequency inverse document frequency) representation, where we use the support vector machine and neural network algorithms for the classification task. We conducted three sets of experiments on a data set consisting of clinical cardiovascular notes from the Netherlands to test the sentence classification performance of the proposed method in comparison with prevalent approaches. The results show that the proposed ETM approach outperformed state-of-the-art baselines.

2.1 Introduction

In recent years, with the development of intelligent information systems for electronic health records (EHRs) inferring patterns, topics, and knowledge from large-scale clinical textual data has emerged as an important and challenging task for a wide range of healthcare applications, such as the classification of disease history, event prediction, topic detection, and patient identity anonymization. While using free text in EHR is useful for medical practitioners, it poses technical challenges for text mining and natural language processing (NLP) (Demner-Fushman, Chapman, & McDonald, 2009; Sevenster, Bozeman, Cowhy, & Trost, 2015; Jonnagaddala et al., 2015; Ghassemi et al., 2014). Some challenges in this area are short sentences, inconsistent structure between texts, unstructured texts, abbreviations, and errors of spelling and grammar. In light of the above, there is a need for tools for automatic text mining to extract implicit, previously unknown, and useful information from data. This study proposes a text mining model for patients' disease history detection, where the records are sentences and labels are binary values that show the presence of disease history.

Many researchers have examined the task of mining clinical text for applications in healthcare (Demner-Fushman et al., 2009; Sevenster et al., 2015; C. Friedman, Shagina, Lussier, & Hripcsak, 2004; Byrd et al., 2014; Torii et al., 2015; Khalifa & Meystre, 2015; Kozlowski & Rybinski, 2019; Shen et al., 2018) and have approached it as a basic text classification problem. Two major challenges in clinical

text classification are the unstructured and short representations of text. Short texts refer to texts with limited context, where the sparsity of patterns of word co-occurrence in the content makes text mining difficult (Zelikovitz & Hirsh, 2000; Sriram, Fuhry, Demir, Ferhatosmanoglu, & Demirbas, 2010; Cheng, Yan, Lan, & Guo, 2014; Yin, Shi, & Wang, 2017; Mirończuk & Protasiewicz, 2018; Unnikrishnan, Govindan, & Kumar, 2019). Medical sentences are example of short texts, where the very small number of words in one medical sentence in EHR texts leads to a large classification error (Zelikovitz & Hirsh, 2000; Unnikrishnan et al., 2019). In clinical text mining, the problem of short text classification is often disregarded (S. Cao et al., 2017). Studies on short text classification for EHR data are mainly based on external dictionaries (ontologies) created by medical experts. In practice, we often do not have dictionaries or do not know in advance of ontologies that might be relevant to the specific domain of application. In addition, for clinical prediction, model interpretability helps to understand the distribution of outcome based on the input words. Therefore, there is increasing demand for automated explainable tools that can analyze and classify EHR free texts. In this study, with the aim of extracting sentences containing medical history from EHR texts, we propose the ETM (enrichment by topic modeling) algorithm for automatic sentence classification for clinical notes. The novelty of the ETM is in the underlying clustering approach that extracts related knowledge from the data set without the need for external dictionaries, such as a medical ontology, to tackle the sparsity of patterns of word co-occurrence. The proposed clustering algorithm is based on latent Dirichlet allocation (LDA) algorithm (Blei et al., 2003), and uses a dynamic weighting mechanism to enrich the data. This algorithm first clusters the initial data set of clinical notes to generate the distribution of hidden topics in clinical notes and probabilities of words in the given topic. Subsequently, the proposed weighting mechanism assigns a weight to every text in terms of its length to mitigate the sampling error inherent in sparse texts by interpolating between the observed word counts and the implied number obtained from an unsupervised model. The proposed ETM yields a smoothed data set that balances individual observations with generic patterns extracted by the LDA algorithm to improve sentence classification.

This study uses clinical notes from a data set collected by the department of Cardiology of the University Medical Center Utrecht (UMCU). The UMCU EHRs encompass free text fields in which different short clinical texts can be entered: e.g. patient anamnesis, physical examination, and medical history. Patients' disease history detection is an example of one classification task on texts from UMCU clinical notes. In this study, each short text is considered one sentence; using a list of delimiters in the experiments. Medical personnel at the department of Cardiology of the UMCU have requested such a system to help them understand past cases from EHR records with similar histories to present ones. Given the nature of free texts, sentence classification is necessary as the first step to extract

the disease history, where not all medical history is clearly delineated and may be provided in free texts at the discretion of the physician.

Thus, this study contributes to the field in the following ways: (i) It presents a method for automatic sentence classification for clinical notes to tackle the problem of the sparsity of patterns of word co-occurrence. (ii) It uses the output of the clustering algorithm as an internal source for enriching short sentences. (iii) It uses the composition of the topic–word distributions and topic distributions of a document with the interpretable TFiDF (term frequency–inverse document frequency) representation. (iv) It takes the shortness of the text into account for enriched representation.

The remainder of this paper is structured as follows: Section 2.2 gives an overview of related work on clinical text classification, sentence classification and short text classification. In Section 2.3, we introduce the proposed ETM approach. Section 2.4 presents an intuitive explanation of the proposed unsupervised model-based smoothing idea, and Section 2.5 details experimental evaluations of the proposed method and a discussion of the results. It shows the usefulness of the proposed method for clinical sentence classification. Finally, in Section 2.6, we offer concluding remarks and directions for future research.

2.2 Related Work

2.2.1 Clinical Text Classification

EHR data contain a large amount of text in which useful patterns need to be automatically identified. Machine learning and text mining algorithms with different data representation methods have been used to study the classification of clinical notes. Mujtaba et al. (2019) presented a comprehensive review of articles on clinical text classification published in 2013–2018. Based on their study, the most extensively employed clinical texts for classification are pathology reports, radiology reports, and Medline biomedical documents. In a majority of studies, the bag-of-words (BOW) representations: binary, term frequency, and TFiDF feature representations were determined to be beneficial. A significant number of the studies have used either supervised machine learning or rule-based approaches.

Many approaches to clinical text classification rely on medical ontologies (dictionaries), such as the unified medical language system (UMLS) meta-thesaurus, and medical subject headings (MeSH), to glean knowledge from clinical notes. Yao et al. (2019) proposed an approach that combines rule-based features and a knowledge-guided convolutional neural network for effective disease classification. They used concepts from the UMLS meta-thesaurus. Similarly, Kocbek et al. (2016) combined three clinical reports—from pathology, radiology, and patients' admission-related meta-data—and used a support vector machine (SVM) with a bag-of-phrases from the UMLS meta-thesaurus to predict the rate of admissions

against disease.

On the contrary, some clinical text classification studies have used non-dictionary-based approaches instead of dictionary-based methods. For instance, Bui and Zeng-Treitler (2014) applied regular expressions to extract snippets of text from clinical notes containing specific words and built an SVM classifier to categorize them. Fodeh et al. (2018) used unstructured text narratives in the EHR to derive pain assessments from clinical notes on patients with chronic pain. They developed their system based on different machine learning classifiers, among which random forest achieved the best results. Blanco, Casillas, Pérez, and de Ilarraza (2019) used several deep learning classification models for assigning multiple ICD codes to clinical documents. They implemented binary logistic regression, a neural network with three fully connected hidden layers, and a bidirectional gated recurrent unit for text classification.

Nevertheless, the problem of clinical sentence classification was not covered in work by Mujtaba et al. (2019), because a few studies have sought to derive the knowledge hidden in clinical short notes (C. Friedman et al., 2004; S. Cao et al., 2017; Mujtaba et al., 2019; Hughes, Li, Kotoulas, & Suzumura, 2017; Lv, Deng, Liu, Cui, & Lu, 2016). Hughes et al. (2017) applied convolutional neural networks (CNNs) with a distributed word representation to medical text classification at the sentence level. They evaluated the learning of complex data representations using the algorithm instead of feature engineering for clinical knowledge representation. Lv et al. (2016) used sentence segmentation, word segmentation, part of speech and entity extraction for text preprocessing to extract features for short text classification in EHRs. In their approach, TFiDF and latent semantic analysis are used to select features that represent the vocabulary for short text classification from several entity dictionaries. In addition, a dependency parser is applied to texts where the dependency relations are used as features for text classification. S. Cao et al. (2017) proposed a knowledge-guided short text classification system for healthcare applications, and claimed that text in the healthcare domain contains domain-specific or infrequently appearing words that can lead to poor embedding owing to a lack of training data. They proposed a bidirectional long short-term memory deep neural network to perform short text classification tasks. Their approach is a domain knowledge-guided attention model that uses the domain dictionary at hand to refine classification performance.

The main difference between the above studies on clinical text classification and our approach is that the former studies used domain dictionaries and disregarded the unlabeled data. Our approach uses the unlabeled data for the unsupervised model-based smoothing, and deploys the labeled data for the sentence classification model.

2.2.2 Sentence Classification and Short Text Classification

Impressive progress has been made on the problem of text classification, but few studies have tackled sentence classification (Kozlowski & Rybinski, 2019; Zelikovitz & Hirsh, 2000; Cheng et al., 2014; Yin et al., 2017; Khoo, Marom, & Albrecht, 2006; Kim, 2014; Jurafsky & Martin, 2019; Aggarwal, 2018). Unlike the traditional text classification problem, sentence classification pose two main challenges. First, patterns of word co-occurrence are sparse in the feature space, where a sentence contains only several to a dozen words. Second, texts face the challenge of a large-scale and manual labeling task, where with sentences this task is more burdensome as they are very small samples causing to increase noise and reduce classification accuracy.

Several techniques have been proposed to tackle the challenges posed by sentence classification, including dimension reduction (Zelikovitz & Hirsh, 2000; Sriram et al., 2010; Khoo et al., 2006; Bollegala, Atanasov, Maehara, & Kawarabayashi, 2018), topic modeling (Cheng et al., 2014; M. Chen, Jin, & Shen, 2011; Yang, Lu, Yang, Yao, & Wei, 2015), clustering (Kozlowski & Rybinski, 2019; Yin et al., 2017; Bollegala et al., 2018; Dai, Sun, & Liu, 2013; Kozlowski & Rybinski, 2017; Yang, Huang, & Cai, 2019), and word embedding (Kozlowski & Rybinski, 2019; Kim, 2014; Lee & Dernoncourt, 2016; Hill, Cho, & Korhonen, 2016). Kim (2014) proposed a single layer of CNN applied for sentence classification. He concluded that despite little tuning of hyperparameters, unsupervised pre-training of word vectors is an important ingredient in deep learning for sentence classification. Zelikovitz and Hirsh (2000) developed a method to reduce error rates in short text classification by using a combination of labeled training data plus a large body of "uncoordinated background knowledge" that is a secondary corpus of unlabeled but related longer documents. They used the WHIRL method (Cohen, 1998) for text classification, an information integration tool designed to query and integrate varied sources of text from the Web. Sriram et al. (2010) proposed an intuitive approach to classify the short texts in tweets by using author information and features of texts. Yin et al. (2017) proposed a short text classification technique based on a combination of the K-nearest neighbors (KNN) and hierarchical SVM classification. They used KNN to initially group labels of the samples to create subclasses and then they applied a SVM algorithm as a hierarchical multi-class classification to each group to classify labels. Cheng et al. (2014) proposed a biterm topic model to capture topics in short texts based on aggregated biterms in the entire corpus to tackle the sparsity of patterns of word co-occurrence in texts. They defined the biterm as an unordered word pair co-occurring in a short text. They considered the corpus as a mixture of topics, where each biterm is drawn independently from a specific topic. Yang et al. (2015) proposed a topic model to extract key phrases for short text classification using the idea that knowledge incorporation can solve the problem of sparsity. Their ap-

proach extracts topics from texts by focusing on phrases in the generative process of documents. Bollegala et al. (2018) developed ClassiNet, a network of binary classifiers trained to predict missing features from a given short text for text classification. ClassiNets solves the problem of feature sparseness by generalizing word co-occurrence graphs by considering implicit co-occurrences between features. Dai et al. (2013) proposed the Crest to generate topic clusters from training data by exploiting a clustering method. Crest uses topic information to extend the representation of short texts and define a new feature space. It subsequently measures the cosine similarity between a document and clusters as augmented features of the document for classification. Lee and Dernoncourt (2016) presented a model on the basis of recurrent and convolutional neural networks. Their model incorporates preceding short texts for sequential short text classification. This model comprises two parts. The first part generates a vector representation for each text and the second part classifies the vector representations of the current text as well as a few preceding short texts using a two-layer feed-forward neural network. Kozlowski and Rybinski (2017, 2019) used a neural network-based distributional model for enriching the semantic meaning of short texts for clustering. They proposed the SnSRC clustering algorithm that uses the SnS method (Kozlowski & Rybinski, 2017), a knowledge-poor text mining algorithm to sense induction, a language-independent approach. They trained their model using continuous bag-of-words and negative sampling, and computed cosine similarity between the mean vector of the embeddings for the text and the vectors for each word in the distributional model. The retrieved words with the highest semantic similarity were added as additional term features to the initial BOW text representation. In their study, especially in cases involving a specific domain language, the semantic enrichment of texts by applying neural networks improved the quality of clustering. Hill et al. (2016) overcame feature sparseness in sentence representations by embedding them into a low-dimensional, dense space. They compared deep neural language models that compute sentence representations from unlabeled data with prevalent methods for word representation, and concluded that the unsupervised BOW models delivered the best performance in terms of sentence representation compared with supervised ones.

Current methods for sentence classification and short text classification either represent texts in a lower-dimensional space to reduce feature sparseness or add data to the text to enhance the quality of the feature space. The main outstanding challenge is the construction of external knowledge repositories, a labor-intensive task in applications of domain-specific clinical text mining. We propose an approach to tackle this challenge in clinical sentence classification that deploys an unsupervised scheme for enriching the original data set by internal knowledge acquisition, where the length of each document is considered by a dynamic weighting mechanism. The proposed approach uses the output of the unsupervised scheme as an internal source for enriching that does not employ any external dictionary.

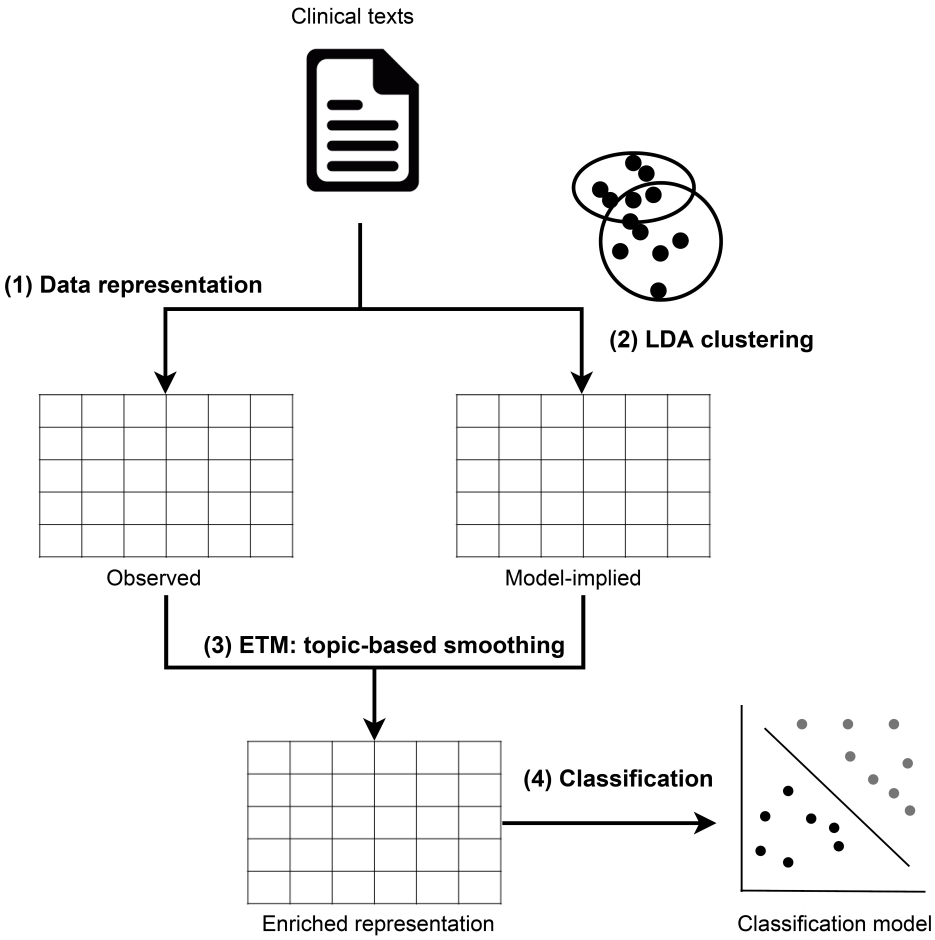


Figure 2.1: The clinical sentence classification model.

2.3 Proposed Methodology

The model for clinical sentence classification proposed in this study is shown in Figure 2.1. This model consists of the following four steps.

- **Data representation,** i.e., preprocessing of clinical texts consisting of sentence detection and extraction, tokenization, spell correction, and representation.
- **LDA clustering,** i.e., using the LDA topic model to cluster sentences in collections of documents to obtain the probabilities of the distributions of document–topic and topic–word in the data set.

- **ETM: Topic-based smoothing,** i.e., using the ETM algorithm as a smoothing method to enrich the representation of clinical sentences according to distribution probabilities of the LDA model.
- **Classification,** i.e., using machine learning classifiers to classify enriched texts. The classification algorithms used in this model are discussed in the experiments' section.

2.3.1 Data Representation

DEDUCE (Menger et al., 2018), a pattern matching tool, is used for automatic de-identification of Dutch medical texts, to anonymize clinical notes for legal and privacy reasons. De-identification process removes patients-level private data that comprise names of patients, names and identification of nurses and doctors, addresses and dates. All texts are then tokenized using *NLTK* library (Bird, Klein, & Loper, 2009) and the *Python scikit-learn* (Pedregosa et al., 2011) feature extractor. *NLTK* sentence tokenizer uses an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences; and then uses that model to find sentence boundaries. This approach has been shown to work well for many European languages (Bird et al., 2009). The punctuation marks that are used to separate sentences in our case study are period, question mark, exclamation point, and semicolon. To handle spelling errors in Dutch texts, the *Python* package *language-check* [1] is used, which is a wrapper for the *LanguageTool*[2] package. *LanguageTool* is an open source proofreading software that can detect and correct spelling errors in more than 20 different languages. The effect of the spell-checker on sentence classification is not evaluated in this study and will be discussed in detail in future work.

Each clinical sentence (document) from the data set is represented by a normalized V-dimensional vector weighted by the TFiDF measure. TFiDF is a BOW representation model that stands for term frequency—inverse document frequency, and is defined as follows:

$$\begin{aligned} \mathrm{tf}_{d,i} &= \frac{n_{d,i}}{\sum_v n_{d,v}} \\ \mathrm{idf}_i &= \log \frac{|C|}{|C_i|} \\ \mathrm{TFiDF}_{d,i} &= \mathrm{tf}_{d,i} \times \mathrm{idf}_i \end{aligned} \tag{2.1}$$

where V is the size of the vocabulary, $n_{d,i}$ denotes the number of times the ith word appears in the dth document, $|C|$ denotes the total number of documents

[1] https://github.com/myint/language-check
[2] https://languagetool.org

in the data set, and $|C_i|$ is the number of documents containing the ith word. TFiDF evaluates how important a word is to a document in a data set, where the importance increases proportionally to the number of times a word appears in the document but is offset by the document frequency of the word in the data set. Thus, with this representation, each document in the data set can be regarded as a multinomial distribution over V words, and each dimension reflects the semantic coherence between the ith word and dth document.

2.3.2 LDA Clustering

Topic modeling is a way of discovering topics in unlabeled text data (Cheng et al., 2014; Blei et al., 2003). The LDA is a generative topic model that represents documents as a mixture of topics and assigns certain probabilities to the words. In other words, the LDA model is an unsupervised learning method that seeks patterns by inferring hidden variables in texts by treating words as observations.

Given a document in the form of $(w_1, w_2, ..., w_N)$, and K asked-for topics, the LDA model estimates parameters θ and β. θ is the distribution of hidden topics in each document. β is the probability of each word given the topic. Figure 2.2 shows a graphical representation of the LDA model (Blei et al., 2003), where the nodes are random variables and the edges indicate the conditional dependencies between them. The shaded and unshaded variables indicate observed and latent (i.e., unobserved, hidden) variables, respectively, while the plates refer to repetitions of the steps of sampling with the variable in the lower-right corner referring to the number of samples. As Figure 2.2 shows: the parameter α is a data set-level Dirichlet prior that can be interpreted as the prior number of observations of a topic being sampled in a document before having observed any words from the document. Similarly, parameter η is a data set-level Dirichlet prior that can be interpreted as the number of prior observations of words sampled from a topic before any word from the data set is observed. These two parameters are assumed to be sampled once in the LDA model when generating a data set of documents.

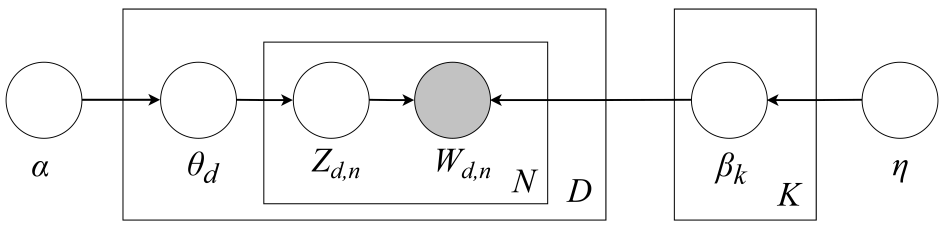


Figure 2.2: The graphical representation of the LDA model.

The variable θ_d is a document-level variable that is sampled once for each document. θ_d is the distribution of hidden topics in the dth document based on

a multinomial distribution with the Dirichlet parameter α. The variable β_k is a topic-level variable that represents the probability distribution of words in topic k. Variables $Z_{d,n}$ and $W_{d,n}$ are word-level variables that are sampled once for each word in each document ($n \in \{1, 2, ..., N\}$). The variable $Z_{d,n}$ is a topic generated by a multinomial distribution with the parameter θ, and variable $W_{d,n}$ is a word sampled from the multinomial distribution with parameters β and Z.

The process of the LDA-clustering algorithm (Blei et al., 2003) implies a joint distribution over the latent and observed random variables (W, Z, β, θ) defined as follows:

$$p(W, Z, \beta, \theta|\alpha, \eta) = \prod_{k=1}^{K} p(\beta_k|\eta) \times \prod_{d=1}^{D} p(\theta_d|\alpha) \left(\prod_{n=1}^{N} p(Z_{d,n}|\theta_d) p(W_{d,n}|Z_{d,n}, \beta_{d,k}) \right) \tag{2.2}$$

Standard statistical techniques can be used to invert the generative process of the LDA model, thus inferring the set of topics responsible for generating a collection of documents. To use the LDA, the key inferential problem to solve is that of computing the posterior distribution of the hidden random variables given the observed words in a document. This posterior distribution is defined in Equation 2.3.

$$p(Z, \beta, \theta|W, \alpha, \eta) = \frac{p(W, Z, \beta, \theta|\alpha, \eta)}{p(W|\alpha, \eta)} \tag{2.3}$$

$$p(W|\alpha, \eta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n \in Z} p(z_n|\theta) p(w_n|z_n, \beta) p(\beta|\eta) \right) d\theta \tag{2.4}$$

This posterior distribution is intractable to compute, and thus approximate inference algorithms are needed for the posterior estimations of β, θ, and Z. The most common approaches used for making inferences in the LDA model are expectation maximization, Gibbs sampling, and variational inference.

2.3.3 ETM: Topic-based Smoothing

Sentence classification is different from traditional text classification in the brevity of the text involved. A solution to improve classification is to enrich the data representation of sentences before training a machine learning model.

Two main approaches have been used to enrich the representation of sentences. The first is to obtain the contextual information of sentence and add more data, and the second approach involves uncovering latent topics from a data set and adding topic-related information to smoothen the representation of sentences. We combine the ideas underlying these two approaches by introducing the ETM algorithm as a topic-based smoothing method. To enrich the feature space of a

sentence, the ETM algorithm matches the inferred probability distributions from the LDA model to words of sentences. sentences are represented as a TFiDF matrix, a $N \times V$ matrix where the rows denote the texts and the columns contain TFiDF values of the chosen words. Then, by applying a method of inference, the ETM extracts the topic distributions and topic assignments for the TFiDF matrix of texts.

The ETM exploits topic analysis to enhance features in sentences by assigning weights to words on the basis of the topics of inference, as the internal features of texts. This approach is applied to clinical notes to improve classification performance. When dealing with sentences, especially when manual labeling is labor intensive, the ETM can use unlabeled data to enrich the quality of the available labeled data. The overall procedure of the ETM is outlined in Algorithm 2.1.

Algorithm 2.1: ETM algorithm

```
1  Input: θ, β, M
2  Output: C
3  for each document in d = {1, 2, ..., D} do
4      Calculate γ_d by Equation 2.5;
5      for each word in i = {1, 2, ..., N} do
6          ω = 0
7          for each topic in k = {1, 2, ..., K} do
8              ω = ω + θ_{d,k} β_{k,i}
9          end
10         C_{d,i} = M_{d,i} + ω γ_d
11     end
12 end
```

In this algorithm, M is the TFiDF matrix and C is the enriched matrix. D is the number of documents in the data set, K is the number of topic clusters, and N is the number of words in the vocabulary. For each sentence, the ETM computes a dynamic enrichment weight γ as in Equation 2.5:

$$\gamma_d = \frac{m}{n_d} \tag{2.5}$$

m is the average length of the sentences and n_d is the number of words in the text document indexed by d. The weight γ is computed to consider the length of each sentence in the enrichment with the ETM. This means that if a text is longer than the average, the weight of the enrichment decreases. The ETM calculates ω as in Equation 2.6, where ω is the enrichment value when information on the distributions θ and β is available. The ETM considers the original representations using β as the posterior probability of each word given the topic, and θ is the

posterior distribution of hidden topics in each document. The ETM updates the representation by adding the enrichment value ω as in Equation 2.6.

$$\begin{aligned} \omega_{d,i} &= \sum_{d=1}^{D}\sum_{i=1}^{N}\sum_{k=1}^{K}\theta_{d,k}\beta_{k,i} \\ C_{d,i} &= M_{d,i} + \sum_{d=1}^{D}\sum_{i=1}^{N}\omega_{d,i}\gamma_d \end{aligned} \tag{2.6}$$

The ETM algorithm enriches each document of the data set by incorporating the length of the document, the posterior distribution of hidden topics in the document, the probabilities of each word given the topic, and the value of the TFiDF of the word. This algorithm considers the length of each document as it incorporates an enrichment dynamic weight with greater values for shorter documents. The idea of considering the length of a text in the enrichment process is as in empirical data sets, where some sentences are long enough while others are short and need to be enriched. The ETM assumes that each document is a mixture of corpus-wide topics, and gains internal knowledge by taking advantage of the patterns of clustering. These patterns contain more contextual information on sentences that can improve clinical sentence classification performance. Because the ETM algorithm uses internal knowledge of the data set as the main source of enrichment, its effectiveness is data dependent.

2.4 Intuitive Explanation

The intuition behind using a clustering method is that, in the BOW representation, sentences are simply very small samples from an underlying multinomial distribution: in this situation, smoothing should present a favorable bias-variance tradeoff, particularly if the smoothing is done towards a latent representation correlated with the outcome. Figure 2.3 illustrates this intuition [3].

Panel A shows a highly simplified representation of documents as *hypothetical* coordinates in the simplex formed by the true proportion of the words "hypertension" and "complaint" in each document. A hypothetical decision boundary for a binary outcome is also shown. Panel B shows the effect of observing only sentences: each point is a sample from the binomial sampling distribution $\hat{\pi}_i \sim \mathcal{N}[\pi_i, \pi_i(1-\pi_i)/n]$, where the number of words is taken to be small, $n = 10$. The unobservable true points, π_i, are shown as gray crosses. Due to the noise incurred from small sample size, many points are on the wrong side of the decision boundary. Panel C demonstrates the effect of using the proposed ETM algorithm, which consists of estimating topic centroids (crosses in panel C) and smoothing the

[3] Firatheme version 0.2.1 (van Kesteren, 2020)

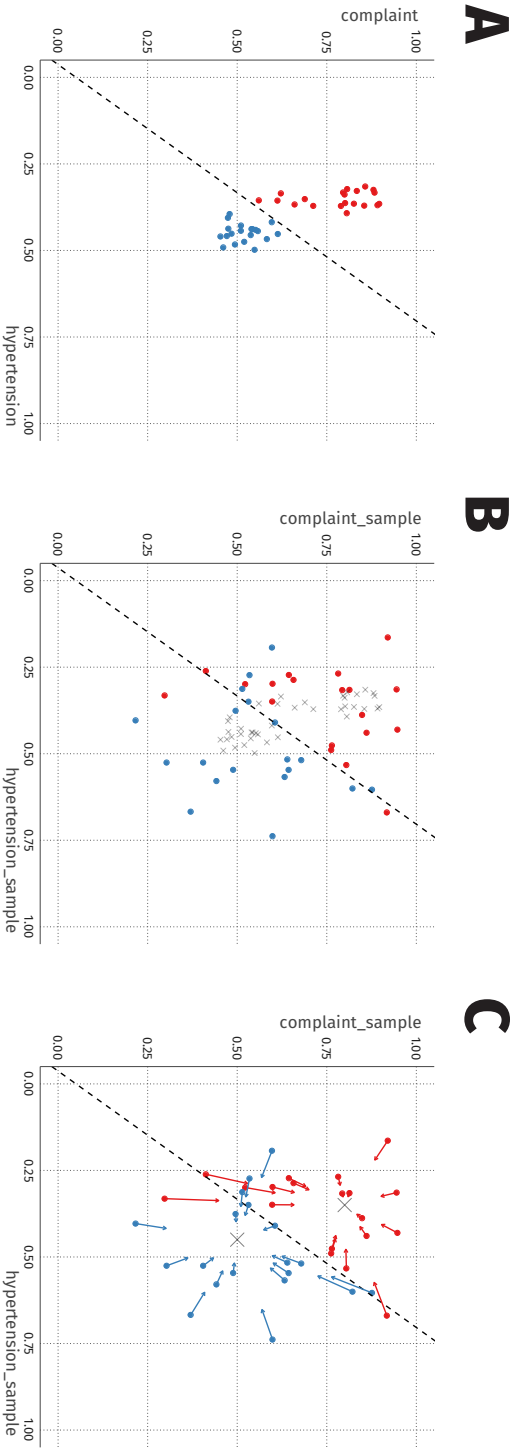


Figure 2.3: Intuition behind the proposed algorithm. Panel A: hypothetical documents, represented as coordinates in a simplex, separated by a decision boundary. Panel B: sentences are samples from the original simplex with small number of words ($n = 10$), increasing noise and reducing classification accuracy. Panel C: Clustering shrinks each observed coordinate to an estimated topic centroid, improving classification accuracy.

observed coordinates towards these estimated centroids. For simplicity of illustration, here smoothing has been performed as $\tilde{\pi}_i = (1-\alpha)\hat{\pi}_i + \alpha\hat{\pi}^*_{k_i}$, where $\hat{\pi}^*_{k_i}$ is the coordinate of the centroid to which point i is estimated to belong, and the amount of smoothing is taken as $\alpha = 0.3$. As can be seen in Figure 2.3, the smoothing (1) reduces the variance of the estimates $\tilde{\pi}_i$, and (2) tends to take misclassified points back across the classification boundary, improving accuracy.

2.5 Evaluation Experiment

In this section, we present the results of clinical sentence classification using several classification algorithms. We evaluated the proposed approach from three aspects. First, we compared the ETM, using different numbers of topic clusters, with the original representations of sentences. Second, we ran experiments using unlabeled data. Third, we compared the ETM with two recently developed methods: Crest (Dai et al., 2013) from a short text classification study, and a CNN-based approach (Hughes et al., 2017) from a medical text classification study.

2.5.1 Data

The UMCU is one of the largest university hospitals in the Netherlands that provides specialized cardiac care. Given the structure of its EHRs, the data are available on a research data platform and can be extracted accordingly. The textual data set used in this study consisted of all clinical cardiovascular notes from doctors or physicians' assistants between 2014 and 2018. A total of 1,002 clinical notes were manually annotated for medical history based on the International Classification of Diseases (ICD10)[4] criteria, and were checked sample wise by doctors. The words in the clinical notes on which the annotation was based were also marked for text mining. These words determine the category of sentences in our data set describing medical history. The description of the data set is provided in Table 2.1. The train and unlabeled data contained 11,053 and 20,200 sentences, respectively. Sentences in the train data were labeled as two classes: with and without medical history. A total of 3,560 records had medical history and 7,493 records were labeled as without medical history.

2.5.2 Example

We present an example (Figure 2.4) to demonstrate the first three steps of the clinical sentence classification model, in this study. This example is used to describe the idea of how text representation could be enriched by incorporating probability distributions from the LDA clustering algorithm.

[4]World Heart Organization, International Classification of Diseases: http://www.who.int

Table 2.1: Data set description.

Category	Number of sentences	Average number of words
Labeled: medical history	3,560	16.51
Labeled: no medical history	7,493	8.76
Unlabeled	20,200	11.19

Data:
S^1: Patient complained of acute chest pain.
S^2: Patient feels absolutely fine.
S^3: She has hypertension.
S^4: Patient feels fine.
S^5: Patient has hypertension and chest tightness.

1) Data representation:
M matrix:

	absolutely	acute	and	chest	complained	feels	fine	has	hypertension	of	pain	patient	she	tightness
S^1	0	0.449	0	0.362	0.449	0	0	0	0	0.449	0.449	0.253	0	0
S^2	0.618	0	0	0	0	0.499	0.499	0	0	0	0	0.348	0	0
S^3	0	0	0	0	0	0	0	0.532	0.532	0	0	0	0.659	0
S^4	0	0	0	0	0	0.634	0.634	0	0	0	0	0.443	0	0
S^5	0	0	0.484	0.390	0	0	0	0.390	0.390	0	0	0.273	0	0.484

2) LDA clustering

Probability distribution of words per topic (β):

	absolutely	acute	and	chest	complained	feels	fine	has	hypertension	of	pain	patient	she	tightness
T^1	0.001	0.111	0.111	0.001	0.001	0.001	0.001	0.220	0.220	0.111	0.001	0.001	0.111	0.111
T^2	0.077	0	0	0.153	0.077	0.153	0.153	0	0	0	0.077	0.305	0	0

Probability distribution of topics per document (θ):

	T^1	T^2
S^1	**0.339**	0.661
S^2	0.024	**0.976**
S^3	**0.969**	0.031
S^4	0.031	**0.969**
S^5	**0.661**	0.339

Enrichment weight:

	γ
S^1	0.73
S^2	1.1
S^3	1.46
S^4	1.46
S^5	0.73

3) ETM: topic-based smoothing

C matrix:

	absolutely	acute	and	chest	complained	feels	fine	has	hypertension	of	pain	patient	she	tightness
S^1	0.038	0.477	0.028	0.436	0.486	0.074	0.074	0.055	0.055	0.476	0.486	0.401	0.028	0.028
S^2	0.700	0.004	0.004	0.164	0.083	0.663	0.663	0.007	0.007	0.004	0.083	0.676	0.004	0.004
S^3	0.005	0.157	0.157	0.009	0.005	0.009	0.009	0.844	0.844	0.157	0.005	0.016	0.816	0.157
S^4	0.109	0.006	0.006	0.217	0.109	0.851	0.851	0.011	0.011	0.006	0.109	0.876	0.006	0.006
S^5	0.020	0.054	0.538	0.429	0.020	0.039	0.039	0.497	0.497	0.054	0.020	0.349	0.054	0.538

Figure 2.4: An example for the ETM enrichment representation showing the first three steps of the sentence classification model.

Data provided in this example contain five sentences and 14 unique words. M is the initial BOW representation, the $TFiDF$ matrix and C is the output of the ETM algorithm, the enriched matrix. The LDA model was applied on the data set to learn two clusters of words (topics). β represents the probability distribution of words for the topics T^1 and T^2. θ represents the probability distribution of

the topics T^1 and T^2 per sentence (document) S^1 to S^5. As shown in Figure 2.4 the ETM algorithm first calculates an enrichment weight (γ) for each sentence in terms of its length. Subsequently, the C matrix is calculated using the M matrix, γ and the clustering outputs: θ and β. The ETM algorithm creates a smoothed data set that balances initial observations with patterns extracted by the LDA algorithm.

2.5.3 Classification

We used an SVM and a multi-layer neural network (NN) as classification algorithms. In the definition of a learning classifier, the training data were the set of documents and the classes were medical history versus no medical history. The objective of the SVM was to find a hyperplane in a high-dimensional feature space that distinctly classified the input data set. By internally employing a kernel trick, it selected the discriminative hyperplane based on the computed support vectors. We used the SVM algorithm with the default parameter settings in *scikit-learn* [5]. The NN classifier in our experiments used a feed-forward architecture and learned to map the input data to the output labels through a series of nonlinear compositions. For sentence classification, the ReLU activation function along with the Adam solver with two hidden layers of 100 units were used. Compared with other non-linearities, the ReLU activation function learns more quickly in deep architectures with many hidden layers. For the learning of the classification algorithms, we chose 80% of the data set as the training set and used the remaining for testing.

2.5.4 Evaluation Measures

To compare the performance of the classifiers, accuracy, precision, recall, and the F1 score were used as the evaluation measures. Precision and recall are useful measures when classes are imbalanced. Precision is a measure of the relevance of the result while recall shows how many truly relevant results were returned. The F1 score is the harmonic mean of precision and recall. These evaluation measures were computed as follows:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{total number of documents}} \tag{2.7}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{2.8}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{2.9}$$

[5] https://scikit-learn.org/stable/modules/svm.html

$$\text{F1 score} = \frac{2\ .\ \text{Precision}\ .\ \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.10}$$

2.5.5 Experiments

2.5.5.1 Classification Performance

We compared the enrichment in representation obtained by the ETM algorithm with the original representation of sentences (denoted by "Raw") using different numbers of topic clusters. Five, 10, 20, and 50 topic clusters were used. For the ETM, we set *n_topics* as the number of topics, $\alpha = \frac{50}{n\_topics}$, $\beta = 0.01$, and the number of iterations = 1000. Figure 2.5 illustrates the accuracy of the ETM approach on clinical sentence classification. The best accuracy value for the representation of Raw was 87.10% using the NN classifier. The ETM outperformed the other methods on the representation when it used more than 10 topic clusters. Using SVM with 50 topic clusters slightly improved the representation of Raw with an accuracy of 87.27%. The highest difference between the representation of Raw and that of the ETM method occurred when the NN classifier was used with 10 topic clusters. This difference was approximately 2.3%.

As is shown in Figure 2.5, with the same number of topic clusters, NN moderately improved the SVM classifier. The highest accuracy using the NN classifier was 89.40% for *n_topics* = 10, and the highest accuracy using the SVM classifier was 87.72% for the same number of topic clusters. Increasing the number of clusters to 50 did not improve classification performance. Using 10 to 20 clusters yielded the best results on our data set in terms of accuracy.

Table 2.2 shows the results in terms of macro-average precision, recall and F1 score to compare the performance of the SVM and NN algorithms on clinical sentence classification.

Table 2.2 shows that the ETM approach improved classification performance considerably compared with Raw in terms of precision and recall. By comparing the precision results, we see that results for Raw were better than those of the ETM using the SVM classifier but inferior to those of the ETM using the NN algorithm.

Table 2.2 shows that using 10 clusters in the ETM approach yielded the best performance in terms of recall. When *n_topics* = 10, the SVM and NN classifiers attained recall values of 89.82% and 89.72%, respectively. The NN classifier yielded the best value of 85.79% using the ETM approach with *n_topics* = 5, and the SVM algorithm obtained a highest value of 83.77% using the ETM approach with *n_topics* = 20. For nearly all settings, the results remained fairly stable when the number of topic clusters was increased from five to 10, but a slight decline in classification performance was noted when the number of topic clusters was increased from 20 to 50.

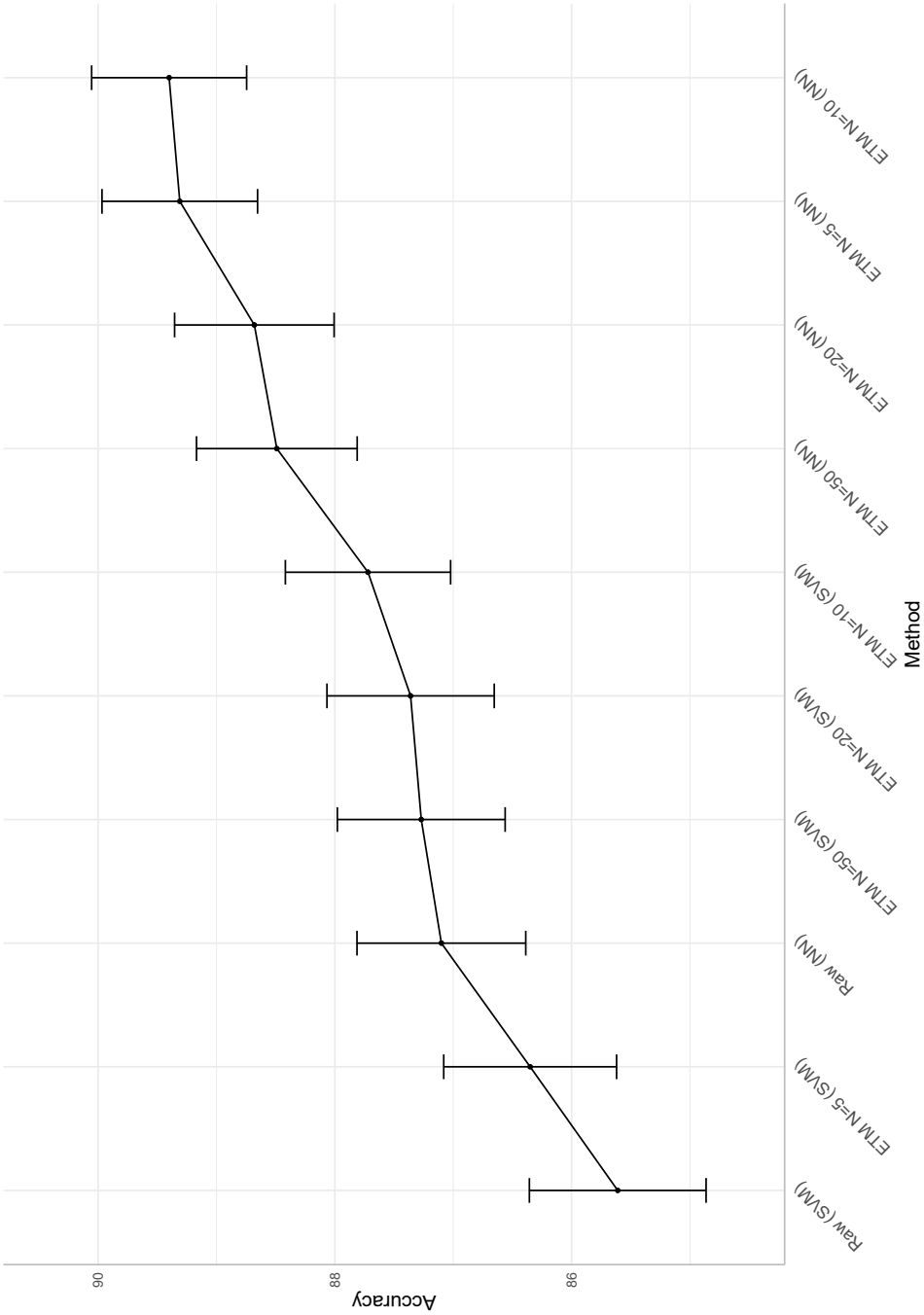


Figure 2.5: Accuracy results of classification performance of Raw methods and the ETM algorithm using N clusters.

Table 2.2: Macro-average precision, recall and F1 score of the assessment of classification performance of Raw methods and the ETM algorithm using N clusters.

Method	Precision	Recall	F1 score
Raw (SVM)	82.21	88.16	85.08
Raw (NN)	82.56	88.47	85.41
ETM $N = 5$ (SVM)	83.00	89.54	86.15
ETM $N = 5$ (NN)	85.79	89.68	87.69
ETM $N = 10$ (SVM)	83.65	89.82	86.63
ETM $N = 10$ (NN)	85.14	89.72	87.37
ETM $N = 20$ (SVM)	83.77	89.57	86.57
ETM $N = 20$ (NN)	84.39	89.06	86.66
ETM $N = 50$ (SVM)	83.02	89.46	86.12
ETM $N = 50$ (NN)	85.01	88.44	86.69

Table 2.3: Precision, recall, and F1 score of the ETM algorithm using the unlabeled data set in addition to the labeled set.

Classifier	Class	Precision	Recall	F1 score
SVM	Medical history	77.12	93.62	84.57
	No medical history	88.43	89.36	88.89
NN	Medical history	82.64	94.09	87.99
	No medical history	89.17	91.32	90.23

These results show that the ETM approach improved classification performance considerably compared with the Raw representation on almost all parameter settings. The ETM approach is robust against changes in the number of topic clusters from five to 20. Even when N was five, the ETM improved the classification performance. This shows its power in enriching representation by using topic clusters.

2.5.5.2 Evaluation Using Unlabeled Data

The previous sets of experiments employed only labeled sets of UMCU data. By using unlabeled data, the ETM can check whether there are words absent from the labeled set. Tables 2.3 and 2.4 show the results of applying the ETM approach using 10 topic clusters on the unlabeled data set in addition to the labeled set.

Table 2.3 shows the results for precision, recall, and the F1 score on the test set for the classes in the data set. The number of sentences with the label *Medical history* in the test set was 576, and was 1635 for the class label *No medical history*. It is notable that the recall values for the label *Medical history* were the highest for both the SVM and the NN classifiers, where the precision values for the label *No*

Table 2.4: Micro- and macro-average precision, recall and F1 score of the ETM algorithm using the unlabeled data set in addition to the labeled set.

Classifier	Metric	Precision	Recall	F1 score
SVM	Micro-avg.	81.97	83.66	82.81
	Macro-avg.	82.78	91.49	86.73
NN	Micro-avg.	86.25	86.47	86.36
	Macro-avg.	85.91	92.71	89.11

medical history were significantly higher than the precision for the *Medical history* class. This might have occurred because the number of texts in the first class label was smaller than in the second class label, and thus the percentage of retrieved relevant sentences was lower than the total number of retrieved texts.

Comparing the results in Table 2.4 with those in Table 2.2 shows the improvement in the performance of the proposed approach obtained by adding the unlabeled set to the labeled set. It is remarkable that the results for recall were higher than those for precision in both classes and for both classifiers. A higher recall than precision means that the classifier tends to extract more of the relevant outputs rather than retrieving correct outputs. As shown in Table 2.2, the highest values for precision and recall without the unlabeled data set were 85.79% and 89.82%, respectively. The former was obtained for the ETM with five clusters using the NN classifier and the latter is for the ETM with 10 clusters using the SVM classifier. The highest macro-average precision and recall of the ETM approach in this experiment were 85.91% and 92.71%, respectively, obtained using the NN classifier with 10 topic clusters. Table 2.4 shows that the NN classifier obtained better results than the SVM classifier. The highest F1 score for the ETM using NN classifier was 90.23% and that for it with the SVM classifier was 88.89%. Both the NN and SVM classifiers were influenced by the enrichment of the data set through the unlabeled data. Thus, when more data are available, the models have a higher chance of improving performance. Moreover, with more data for clinical text classification, the chances of encountering new words in new samples decrease.

2.5.5.3 Comparison Study: Crest, CNN, and ETM

We compared the results of the ETM algorithm with the following two methods:

Crest: Crest generates topic clusters from training data by exploiting a clustering method, and then uses the topic information to extend the representation of short texts (Dai et al., 2013). This approach is similar to that of the ETM as it uses a clustering method. The difference is that Crest uses the cosine similarity between a short text and a topic cluster as similarity vector that is then used for text representation. The ETM does not use a similarity metric but the probability

distributions of documents and topics inferred from the generative process of the LDA. Crest increase the dimensions of the feature space by the number of clusters whereas the ETM uses the same dimensions as the training set.

CNN: As mentioned in Section 2.2, Hughes et al. (2017) implemented a deep CNN model for medical text classification at the sentence level. A CNN model requires that the length of the text have a fixed size as input. Therefore, they chose a maximum word length of 50 for a text, and applied a Word2vec layer of size 100. Their model consisted of two sets of convolutional layers followed by two max pooling layers. They used convolutional filters and applied a dropout function to help prevent overfitting. Then, a fully connected layer with 128 units was followed by a dense layer using a softmax function.

Figure 2.6 shows the results of a comparison among Crest, CNN, and ETM. In the experiments, the CNN model was used with two settings: one experiment used two convolutional layers (*'CNN + 2conv'*) and the other, *'CNN + Word2vec'*, applied the CNN approach using two pairs of convolutional layers followed by two max-pooling layers and dense layers. As shown in the figure, the CNN models had lower accuracy than the Crest and ETM. There are two reasons for this: (1) Feature engineering in the Crest and ETM approaches has been proposed especially for the short text classification problem. (2) The trained word vectors are not rich enough to capture the semantics and diversity in our clinical text collection of Dutch language. While there is no publicly available pre-trained word vectors for Dutch clinical text, Dutch word vectors trained on social media and Wikipedia can be experimented as initial weights for the deep learning models in future work.

The highest accuracy in the experiments on the CNN models using the test set was 79.95% whereas the closest model to this was that of Crest using the SVM classifier, with a value of 86.65%. The models with enriched representations delivered better performance than the CNN classifier, which proves the effectiveness of using smoothing methods to enrich the original representation. The accuracy of Crest using the NN classifier was higher than that of the ETM using the SVM algorithm. The differences between *'Crest + NN'*, and *'ETM(unlabeled) + SVM'* and *'ETM + SVM'* were 1.44% and 0.15%, respectively. This shows the positive effect of using a neural network approach compared with an SVM classifier. In all approaches used in these experiments, the NN classifier had an accuracy of approximately 2% higher than the SVM classifier. The highest accuracy was obtained by the ETM method, 89.64%, when it used the unlabeled data set with the NN classifier. This shows the power of the ETM approach in overcoming the brevity and sparsity of sentences by utilizing topic clusters extracted from the training data for better representation.

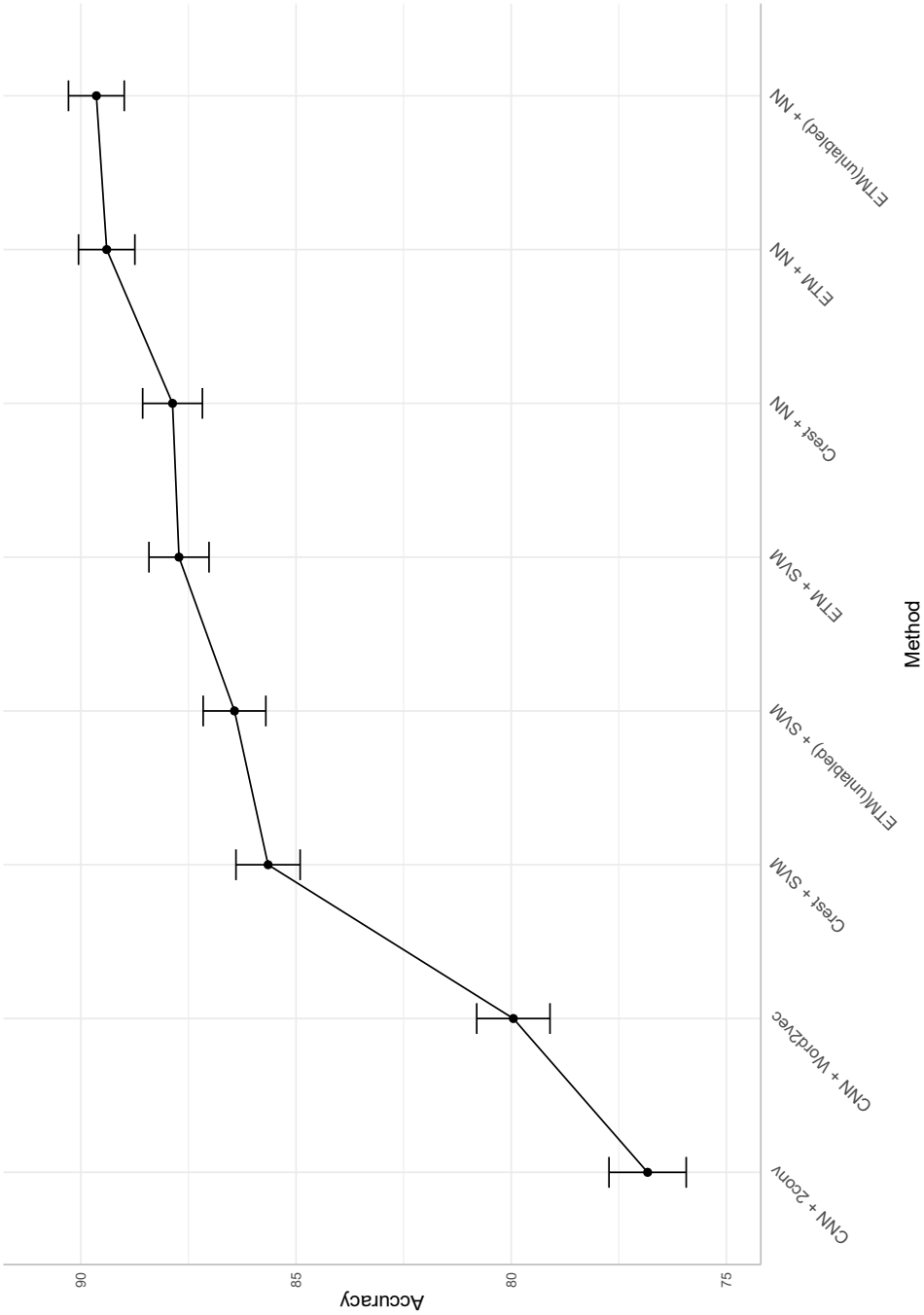


Figure 2.6: Comparison study: Crest, CNN, and ETM.

2.6 Conclusions

EHRs usually store patients' disease history in free-text form. Although this lack of structure might not directly affect patient care in clinical settings, it does affect other uses of the EHR, such as patient recruitment for clinical trials. Automated text analysis using text mining algorithms eliminates administrative burdens and is important for research. The textual classification of clinical sentences is a first step in the automated extraction of medical history. Because of the limited number of words used in clinical sentences, this problem is considered that of short text classification. Current approaches to clinical sentence classification mainly use external dictionaries but this has a number of drawbacks, including the lack of a universal medical dictionary for different languages. This study proposed an unsupervised model-based smoothing method, the ETM approach, that uses an internal knowledge acquisition mechanism without employing any external dictionary. The ETM considers the length of each document in the enrichment phase and adds hidden information behind the topic clusters gained from the clustering algorithm. It is notable that the purpose of the enrichment is to *improve the text classification workflow*; we do not change the original record or the results displayed to a physician. While model interpretability is difficult to achieve in practice, using BOW representation with the ETM approach makes prediction explainable. To mitigate the error in sentence classification, we trained the enriched representation on the SVM and NN classification algorithms, and used clinical cardiovascular notes from the UMCU hospital in the Netherlands. Experimental results showed that applying the proposed ETM approach delivers good classification performance, and is comparable to prevalent alternatives. Moreover, it is simple and easy to implement, where this makes the ETM a promising tool for the analysis of short texts for various applications. With sentences and even short notes that include two or three sentences we may not have enough information to predict heart failure, or to find patients with arrhythmia for clinical trials. In these situations the proposed ETM algorithm, as an internal source for enriching clinical notes, can help to improve accuracy of the disease prediction and the performance in the process of patient recruitment. In future work, we plan to look into the performance of the ETM approach in prognostic prediction models by incorporating other variables from EHRs, and evaluate the efficiency of text mining in creation of clinical trials. Furthermore, we will study the impact of the size of the data set on performance and investigate the use of enriched representations in complex deep learning models.

Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest.

Chapter 3

SALTClass: Classifying Clinical Short Notes using Background Knowledge from Unlabeled Data

Bagheri, A., Oberski, D. L., Sammani, A. Van der Heijden, P. G. M., & Asselbergs, F. W. (2019, in preparation). SALTClass: Classifying Clinical Short Notes using Background Knowledge from Unlabeled Data. https://doi.org/10.1101/801944

Abstract

With the increasing use of unstructured text in electronic health records, extracting useful related information has become a necessity. Text classification can be applied to extract patients' medical history from clinical notes. However, the sparsity in clinical short notes, that is, excessively small word counts in the text, can lead to large classification errors. Previous studies demonstrated that natural language processing (NLP) can be useful in the text classification of clinical outcomes. We propose the software package SALTClass (short and long text classifier) to incorporate the knowledge from unlabeled data, as this may alleviate the problem of short noisy sparse text. The SALTClass package is a machine learning NLP toolkit. It uses seven clustering algorithms, namely, latent

Author contributions: AB and DLO designed the study. AB developed the new software package. AB and AS analyzed the data and experiments. AB wrote the paper. DLO, PvdH and FWA provided feedback on written work.

Dirichlet allocation, K-Means, MiniBatch K-Means, BIRCH, MeanShift, DBScan, and GMM. Smoothing methods are applied to the resulting cluster information to enrich the representation of sparse text. For the subsequent prediction step, SALTClass can be used on either the original document-term matrix or in an enrichment pipeline. To this end, ten different supervised classifiers have also been integrated into SALTClass. We demonstrate the effectiveness of the SALTClass NLP toolkit in the identification of patients' family history in a Dutch clinical cardiovascular text corpus from University Medical Center Utrecht, the Netherlands. Using machine learning algorithms for enriching short text can improve the representation for further applications.
SALTClass can be downloaded as a *Python* package from *Python* Package Index (PyPI) website at `https://pypi.org/project/saltclass` and from GitHub at `https://github.com/bagheria/saltclass`.

3.1 Introduction

A considerable amount of the data stored and documented in electronic health records (EHRs) are in the form of unstructured or semi-structured narrative text (Mehta & Pandit, 2018; Demner-Fushman et al., 2009; Jonnalagadda et al., 2017). EHR text may contain short and noisy data. The telegraphic style of clinical narrative notes, including spelling errors or ambiguous abbreviations and measurements, renders information retrieval difficult (Demner-Fushman et al., 2009; Jonnalagadda et al., 2017; H. Wu et al., 2018). In this context, traditional learning classifiers do not perform satisfactorily because clinical short notes do not provide sufficient word occurrences or shared context information, (Cheng et al., 2014; Sriram et al., 2010). By contrast, text mining and natural language processing (NLP) have become the most widely used big-data analytical techniques in healthcare applications (Mehta & Pandit, 2018).

Among the attempts to develop NLP software for healthcare (Jonnalagadda et al., 2017; Alex et al., 2019; C. Friedman et al., 2004; Savova et al., 2010; Jackson et al., 2018; Torii et al., 2015; Jonnagaddala et al., 2015; Byrd et al., 2014; Weeks et al., 2019; Soysal et al., 2017; Sohn et al., 2014), medical language extraction and encoding (MedLEE) (C. Friedman et al., 2004), clinical text analysis and knowledge extraction system (cTAKES) (Savova et al., 2010), CogStack (Jackson et al., 2018), and CLAMP (Soysal et al., 2017) are prominent examples. MedLEE is an NLP system that can extract information from textual patient reports based on controlled vocabularies. It uses a lexicon to map terms into semantic classes, and a semantic grammar to generate formal representations of sentences. It then performs named entity recognition by dictionary look-up, handles abbreviations using a mapping table, and performs word sense disambiguation based on con-

textual rules. cTAKES is an NLP system for the extraction of information from electronic medical free-text. It processes clinical text by identifying clinical entities such as drugs, diseases, and symptoms. CogStack implements data mining techniques that can search clinical data sources using automated information extraction of medical concepts. It uses NLP annotations to generate a timeline for patient interactions with services. CLAMP is abbreviated for clinical language annotation, modeling, and processing. It is a customizable NLP pipeline achieved good performance on named entity recognition and concept encoding for processing clinical text data. CLAMP's components include sentence boundary detection, tokenizer, part-of-speech tagger, section header identification, abbreviation reorganization and disambiguation, named entity recognizer, and rule engine.

Sparsity in the feature space is a characteristic of clinical text that should be appropriately handled. To this end, studies on short text data classification (Cheng et al., 2014; Sriram et al., 2010; Yin et al., 2017; Dai et al., 2013; Bollegala et al., 2018; M. Chen et al., 2011; Yang et al., 2015) have adopted two major approaches to enrich short text. One is to fetch contextual short text information to add more text directly; this approach is called "dictionary-based method." The other approach, which is called "topic-based method," is to extract latent topics from an existing corpus. These topics are then used as features in further applications. The latter approach may lead to information loss, whereas the former cannot be applied in every domain owing to the lack of standard dictionaries, which is also a problem in EHR text classification.

Zelikovitz and Hirsh (2000) presented a dictionary-based method to reduce error rates in short text classification by using a large body of potentially unrelated background knowledge. They used an information-integration tool for text classification to query and integrate varied textual sources from the Web. Dai et al. (2013) proposed Crest, a topic-based method that generates topic clusters from training data. They used topic information to represent short texts by a new feature space. Crest represents short text documents as augmented features using the cosine similarity between every document and the clusters. Cheng et al. (2014) also proposed a topic-based method for short texts whereby topics are captured based on aggregated biterms in an entire corpus to tackle the sparsity problem. They define a biterm as an unordered word-pair co-occurring in a short text, considering the entire corpus a mixture of topics, where each biterm is drawn from a specific topic independently. Another study on short text classification for medical records is the use of a bidirectional long short-term memory recurrent network by S. Cao et al. (2017), where a knowledge-guided short text classification system was proposed for healthcare applications using domain concept dictionaries. It was claimed that clinical notes contain domain-specific or infrequently appearing words. This may result in a poor embedding owing to the lack of training data.

Yu, Ho, Juan, and Lin (2013) proposed *LibShortText* software, which is an open source tool for short text classification. *LibShortText* is a well-implemented

Python package which provides a capability to change the parameters of the support vector machine algorithm. Yu et al. (2013) have demonstrated that because short texts have more features than records the use of linear kernel for SVM algorithm is ideal.

In the present study, we develop a hybrid technique, which is called "intra-clustering approach," that combines the advantages of both dictionary- and topic-based approaches. An important difference between the proposed technique and that in (S. Cao et al., 2017) is that the latter is a dictionary-based method in which if the domain knowledge dictionary is incomplete, a multi-task model is used to learn the domain knowledge dictionary jointly and perform the classification task, whereas the former completely relies on input data in an unsupervised manner. With the proposed intra-clustering approach, we use clustering algorithms to deploy new features from an internal knowledge acquisition scheme. A notable aspect of the proposed method is that it does not use categories of training examples to enrich the representation of short texts. However, a potential problem is that there are words in the test set that have not occurred in the training set. To overcome this, the proposed intra-clustering approach enhances the representation by incorporating background knowledge from unlabeled data. This allows new words in the test set to be incorporated in representation learning.

In what follows, we present SALTClass (short and long text classifier): a *Python* package that applies the proposed approach to clinical short text classification. We use SALTClass on a data set collected by the Department of Cardiology of University Medical Center Utrecht (UMCU). In the experiments, the goal is to classify clinical sentences from patient notes and letters. The results from this classification can identify whether the clinical letters contain information about a patient's medical history. This classification can be used in two ways: (i) to mine a patient's history from clinical letters and present it to the EHR and (ii) as the first step for further standardization of clinical history (e.g., ICD10 and SNOMED coding).

3.2 SALTClass: A Natural Language Processing-based Package

Short text classification can be defined as follows: given a set of documents with representation D, a label from a set of categories is assigned to each document. As short texts are also characterized by their representation sparsity, D should be optimized so that better performance may be achieved in the analysis of EHR text data.

Figure 3.1 shows the semantic flowchart of the SALTClass NLP toolkit. SALTClass optimizes the representation D for short text classification. This toolkit classifies sentences by cleaning them and using a combination of clustering algorithms

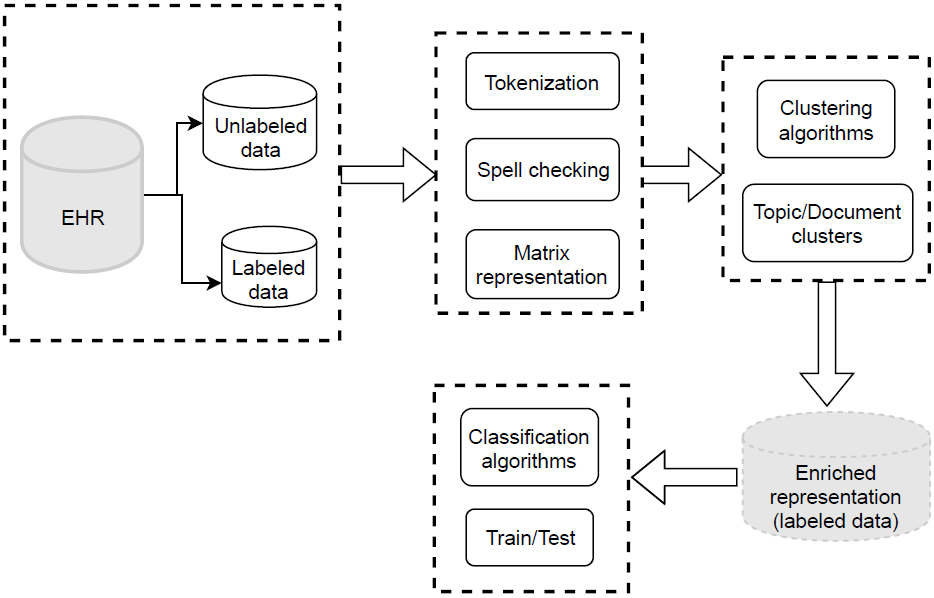


Figure 3.1: Architecture of SALTClass NLP software.

and supervised classifiers. Let D and D^* be the original clinical free-text data set and the enriched data set, respectively. In the framework of SALTClass, D^* is enriched based on related knowledge from word clusters extracted by unsupervised clustering algorithms.

The first step of the SALTClass architecture is preprocessing, which includes detecting and extracting all sentences from D. Sentences will then be split using a tokenization module and will be represented by a vector-space bag-of-words model. The preprocessing step also includes removing spelling errors and stop words from D. The final two steps in the SALTClass architecture are the use of an unsupervised clustering procedure and a supervised classification algorithm.

In SALTClass, the unsupervised intra-clustering procedure is the heart of the architecture. This procedure uses the background knowledge in the data to optimize the vector representation. It pumps cluster information throughout the text body using a smoothing technique. This procedure provides text fragments with additional length. Intra-clustering is a hybrid technique, using the advantages of different modules, including dictionary- and topic-based approaches, smoothing methods, and cluster information. Dictionary-based techniques integrate text data with meta-information from other information sources, such as Wikipedia and WordNet, and topic-based methods represent short text with latent topics from the data set. The intra-clustering algorithm in the proposed NLP toolkit is presented in Algorithm 3.1.

Algorithm 3.1: Intra-clustering algorithm

```
1  Input:
2  D: Document-term matrix
3  C: Matrix of cluster centers
4  m: Mean number of terms per document
5  Output:
6  D*: Enriched document-term matrix
7  for d ∈ 1, ..., n do
8  |  l ← number of nonzero indices in D^d
9  |  γ ← m/l
10 |  for i ∈ 1, ..., v do
11 |  |  D*_i^d ← D_i^d + γC_i^d
12 |  end
13 end
```

The intra-clustering algorithm expects three objects as input: D, C, and m. D is the document-term matrix for the text data set. In this matrix, documents are represented by rows and terms (n-gram words) by columns, and the elements are the counts or the weights. It is not necessary to provide labeled categories with the matrix D; hence, the intra-clustering algorithm uses topic distributions as the background knowledge of the entire data set.

C is the matrix of cluster centers, which is the cluster-term matrix. m is the average number of terms per document in the data set. n is the total number of documents. D_i^d denotes term i in document d of data set D. v is the vocabulary size, that is, the number of unique terms in the data set. l counts the number of individual terms in document d for calculating its enrichment weight γ. The intra-clustering algorithm outputs the enriched representation in the document-term matrix D^*.

Figure 3.2 shows an illustrative example of the intra-clustering algorithm. As can be seen from this example, the proposed intra-clustering method attempts to move documents toward the center of assigned clusters.

This procedure may add words from the center vector to the document vector and change the values of the document vector representation. That is, the proposed algorithm represents short texts by adding information from latent topics to the original annotated data set. This algorithm is an unsupervised learner, and instead of using training data, it uses both training and test examples in choosing the hypothesis of the learner. For classifying clinical short texts, the proposed toolkit allows the unseen words from the test set to be incorporated in the process of representation learning.

With SALTClass, different methods are implemented to enable users to choose a configuration. A summary of the main methods and their parameters in SALT-

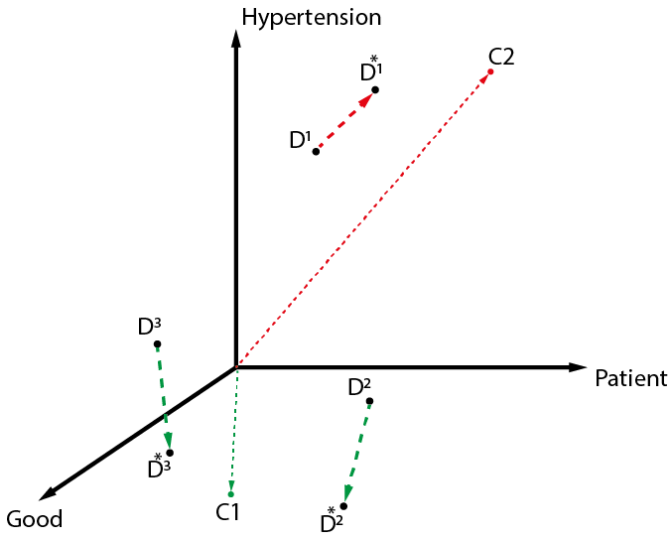


Figure 3.2: Illustrative example of the proposed intra-clustering algorithm. The feature space in this example has three words: Patient, Hypertension, Good. D, C, and D^* denote the original document, the cluster center, and the enriched document, respectively. The documents are enriched toward the center of the cluster to which they belong.

Class is shown in Tables 3.1 and 3.2, respectively.

The list of clustering and classification methods implemented in SALTClass are:

- K-Means
- MiniBatch K-Means
- Latent Dirichlet allocation (LDA)
- Balanced iterative reducing and clustering using hierarchies (BIRCH)
- Density-based spatial clustering of applications with noise (DBSCAN)
- Gaussian-mixture modeling (GMM)
- Meanshift
- Logistic regression (LR)
- Neural networks (NNs)
- K-Nearest neighbors (KNNs)
- Support vector machines (SVMs)
- Decision trees (DTs)

Table 3.1: Overview of methods.

Method	Description	Parameters
SALT	To initialize an object with a training matrix and define a setting	x, y, vocabulary, kwargs
SALT.data_from_dir	To initialize an object from a directory of text files	train_dir, kwargs
initialize_dataset	To read documents from a training path and vectorize them	train_dir, word_vectorizer, language
enrich	To enrich a training set with a method	method, num_clusters, include_unlabeled, unlabeled_dir, unlabeled_matrix
train	To train a supervised classifier	kwargs
predict	To predict a category	data_file

- Random forest (RF)
- AdaBoost (AdaB)
- Gaussian naive Bayes (GaussianNB)
- Multinomial naive Bayes (MultinomialNB)
- Gaussian processes (GPs)

3.3 Experiment Study: Package Evaluation

3.3.1 Example

We present an example to clarify the principle of SALTCLass. In this example, we employ the K-Means and LDA clustering methods from the SALTClass package to illustrate the concept of enrichment. The example has a data set of five sentences (short documents), which are shown in Figure 3.3. This figure also shows the count-vector representations for the short documents. As the vocabulary is small, we do not remove stop words. The average number of terms in the documents is

Table 3.2: Definition of parameters of the methods.

Parameters	Definition
x	Training data: a numerical document-term matrix
y	Target values: categories
vocabulary	Variables (word features) in x
kwargs (SALT method)	Keyword arguments: language="nl", vectorizer="count"
train_dir	Training data: directory folders of text files
kwargs (SALT.data_from_dir method)	Keyword arguments: language="nl", vectorizer="count"
word_vectorizer	Word feature vectorizer: "count", "tfidf"
language	Language of text: "nl", "en"
method	Clustering method: lda, kmeans, mbk, birch, gmm, ms, dbscan
num_clusters	Number of clusters
include_unlabeled	Flag: True, False
unlabeled_dir	Directory folder of unlabeled text files
unlabeled_matrix	Unlabeled document-term matrix
kwargs (training method)	Keyword arguments: classifier="SVM", kernel="poly", degree=2 classifier="SVM", kernel="sigmoid" classifier="SVM", kernel="lin", gamma=2 classifier="KNN", k=3 classifier="DT", depth=5 classifier="RF", depth=5, n_estimators=10, max_features=1 classifier="NN", alpha=1, hidden_layer_sizes=(50,), max_iter=10, solver="sgd", "adam", activation="relu" classifier="AdaB" classifier="GaussianNB" classifier="MultinomialNB" classifier="GP" classifier="LR"
data_file	A text file for prediction

$m = 4.4$. The enrichment weight γ is also shown in Figure 3.3 and is calculated for each document using m and the inverse document length.

Figures 3.4 and 3.5 show the enrichment phase using the K-Means and LDA algorithms, respectively. For intra-clustering, we use the algorithms to generate two document clusters. This can be accomplished in the SALTClass package by using the command *object.enrich(method="kmeans", num_clusters=2)* for the K-Means algorithm. For LDA, we use *object.enrich(method="lda", num_clusters=2)*. The intra-clustering algorithm first calculates the enrichment weight γ for each document. Subsequently, the count vectors are enriched using the clustering outputs. In the K-Means algorithm, the count vectors use the corresponding center vector from the clusters to update their values. In the LDA algorithm, the count vectors use both the document topic and the topic word distributions to modify the representation.

Figure 3.4 also shows the K-Means clustering labels for each document. Figure 3.5 shows the word distributions per topic and the topic distributions per document. It can be seen that the algorithm outputs the matrix D^*. K-Means and LDA clustering generate two different D^* matrices. One noticeable difference between the outputs is the zero and non-zero values for the same cell. This is due to the LDA assumption that documents have multiple topics (Blei et al., 2003).

3.3.2 Data

UMCU is one of the largest university hospitals in the Netherlands that provide specialized cardiac care. Given the structure of their EHRs, data are available in a research data platform and can be extracted accordingly. The textual data set used in this study comprises all clinical cardiovascular notes from medical doctors or physician assistants between 2014 and 2018. 1002 Dutch clinical notes have been manually annotated for medical history, based on the international classification of disease (ICD10) criteria and were sample-wise checked by medical doctors. Conflicts in annotation were resolved through discussion. The words in these clinical notes on which the annotation was based were also marked for text mining purposes. These words delineate sentences that contain medical history. The training data set was generated based on this delineation. After annotation, the training data for short text classification contained 11,053 sentences, where 3,560 of them were related to medical history. Along with the labeled clinical notes, 20,200 unlabeled clinical cardiovascular sentences were used for the experimental study of the proposed NLP software.

3.3.3 Dutch Text Preprocessing

In NLP, a major type of preprocessing is to filter out stop words and spelling errors. Clinical notes may have misspellings and useless words, which are referred

Dataset D:
D^1: She feels fine.
D^2: He feels absolutely fine.
D^3: She has hypertension.
D^4: Patient complained of acute chest pain.
D^5: Patient has hypertension and chest tightness.

	Enrichment weight (γ)
$\mathbf{D^1}$	1.46
$\mathbf{D^2}$	1.1
$\mathbf{D^3}$	1.46
$\mathbf{D^4}$	0.73
$\mathbf{D^5}$	0.73

Matrix representation by count vectorizer:

	absolutely	acute	and	chest	complained	feels	fine	has	he	hypertension	of	pain	patient	she	tightness
$\mathbf{D^1}$	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0
$\mathbf{D^2}$	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0
$\mathbf{D^3}$	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
$\mathbf{D^4}$	0	1	0	1	1	0	0	0	0	0	1	1	1	0	0
$\mathbf{D^5}$	0	0	1	1	0	0	0	1	0	1	0	0	1	0	1

Figure 3.3: Example of enrichment representation. The data are represented with CountVectorizer from the *scikit-learn* package in *Python*.

K-Means cluster centers:

	absolutely	acute	and	chest	complained	feels	fine	has	he	hypertension	of	pain	patient	she	tightness
$\mathbf{C^1}$	0.5	0	0	0	0	1	1	0	0.5	0	0	0	0	0.5	0
$\mathbf{C^2}$	0	0.33	0.33	0.66	0.33	0	0	0.66	0	0.66	0.33	0.33	0.66	0.33	0.33

Cluster label per document:

	cluster label
$\mathbf{D^1}$	1
$\mathbf{D^2}$	1
$\mathbf{D^3}$	2
$\mathbf{D^4}$	2
$\mathbf{D^5}$	2

D matrix:*

	absolutely	acute	and	chest	complained	feels	fine	has	he	hypertension	of	pain	patient	she	tightness
$\mathbf{D^1}$	0.73	0	0	0	0	2.46	2.46	0	0.73	0	0	0	0	1.73	0
$\mathbf{D^2}$	1.55	0	0	0	0	2.1	2.1	0	1.55	0	0	0	0	0.55	0
$\mathbf{D^3}$	0	0.48	0.48	0.97	0.48	0	0	1.97	0	1.97	0.48	0.48	0.97	1.48	0.48
$\mathbf{D^4}$	0	1.24	0.24	1.48	1.24	0	0	0.48	0	0.48	1.24	1.24	1.48	0.24	0.24
$\mathbf{D^5}$	0	0.24	1.24	1.48	0.24	0	0	1.48	0	1.48	0.24	0.24	1.48	0.24	1.24

Figure 3.4: Intra-clustering with K-Means. The K-Means clustering method is applied to the example to generate two document clusters. The matrix D^* contains the enriched representation for the documents.

LDA word distribution per topic:

	absolutely	acute	and	chest	complained	feels	fine	has	he	hypertension	of	pain	patient	she	tightness
T^1	0.001	0.071	0.071	0.142	0.071	0.001	0.001	0.142	0.001	0.142	0.071	0.071	0.142	0.001	0.071
T^2	0.124	0.001	0.001	0.001	0.001	0.247	0.247	0.001	0.124	0.001	0.001	0.001	0.001	0.247	0.001

Topic distribution per document:

	T^1	T^2
D^1	0.031	**0.969**
D^2	0.024	**0.976**
D^3	**0.656**	0.344
D^4	**0.984**	0.016
D^5	**0.984**	0.016

D matrix:*

	absolutely	acute	and	chest	complained	feels	fine	has	he	hypertension	of	pain	patient	she	tightness
D^1	0.176	0.005	0.005	0.008	0.005	1.35	1.35	0.008	0.176	0.008	0.005	0.005	0.008	1.35	0.005
D^2	1.133	0.003	0.003	0.005	0.003	1.265	1.265	0.005	1.133	0.005	0.003	0.003	0.005	0.265	0.003
D^3	0.063	0.069	0.069	0.137	0.069	0.125	0.125	1.137	0.063	1.137	0.069	0.069	0.137	1.125	0.069
D^4	0.002	1.052	0.052	1.103	1.052	0.003	0.003	0.103	0.002	0.103	1.052	1.052	1.103	0.003	0.052
D^5	0.002	0.052	1.052	1.103	0.052	0.003	0.003	1.103	0.002	1.103	0.052	0.052	1.103	0.003	1.052

Figure 3.5: Intra-clustering with LDA topic modeling. The LDA algorithm is applied to the example to generate two document clusters. The matrix D^* contains the enriched representation for the documents.

to as stop-words. These can be removed without any negative consequences to the training model. There is no universal list of stop words because a word can be meaningful or meaningless depending on the context. Making a list of stop words for an EHR is beyond the scope of this study. However, the *NLTK* [1] *Python* package provides lists of stop words in 21 different languages, including a list of Dutch stop words. To handle spelling errors in texts, we used the *Python* package *language-check* [2], which is a wrapper for the *LanguageTool* [3] package. *LanguageTool* is an open source proofreading software that can detect and correct spelling errors in more than 20 different languages. This package is freely available under the *LGPL*2.1 or later. *LanguageTool* functions properly with *Python*3.6 and *JDK*8.

3.3.4 Results

We use precision, recall and F1 score results by 5-fold cross validation in the experiments to evaluate the performance of the SALTClass toolkit. Precision is defined as the fraction of relevant documents among the retrieved documents, whereas recall is the fraction of relevant documents that have been retrieved over the total amount of relevant documents. The F1 score is defined as the harmonic average of the precision and recall of the test (Jurafsky & Martin, 2019).

Table 3.3 shows the F1 score results of the first experiment, where we checked the effectiveness of SALTClass using *count* (term frequency) and *tfidf* (TFiDF) vector representations. Setting1 is the count-vector representation and Setting2 is the TFiDF representation. TFiDF is short for "term frequency inverse document frequency" and is a numerical statistic that is intended to reflect the importance of a word to a document in a collection of documents. Neither Setting1 nor Setting2 calls the *enrich* function. These two settings are coupled in four different experiments with the following supervised classifiers:

- **AdaBoost** is a type of ensemble learning for classification.
- **KNN** is a non-parametric instance-based learning algorithm.
- **NN** learns to map the input data to the output labels through a series of nonlinear compositions.
- **SVM** learns an objective function by employing internally a kernel trick.

As shown in Table 3.3, the results for the classifiers are highly similar in both representations. SVM and NN obtained much improved results with TFiDF. KNN performed better with count representation, and AdaBoost gained similar results with two representations. Nevertheless, the results of the two bag-of-words models

[1] https://www.nltk.org/
[2] https://github.com/myint/language-check
[3] https://languagetool.org

Table 3.3: F1 score results of two representations with the learning classifiers.

Experiment	AdaBoost	KNN	NN	SVM
Setting1	82.87	72.51	84.13	79.49
Setting2	82.92	71.48	85.41	85.08

demonstrate the average superiority of TFiDF over count-vector representation. This is because TFiDF extracts more descriptive features from a document.

Tables 3.4 and 3.5 compare the precision and recall results for three settings, respectively. Setting2 is the TFiDF representation denoted by *Raw*. Setting3 is the TFiDF representation calling the *enrich(num_clusters=10)* function to apply the intra-clustering algorithm. This experiment shows the results of the K-Means and LDA algorithms, as the unsupervised method used in the framework of the intra-clustering algorithm. We chose 10 as the number of clusters based on the experiments on different values. Setting4 is the same as Setting3 assigning *True* to the parameter *include_unlabeled* and a directory path to the folder of unlabeled data in the parameter *unlabeled_dir*.

Table 3.4: Precision results of the learning classifiers.

Method	Experiment	AdaBoost	KNN	NN	SVM
Raw	Setting2	79.64	70.88	82.56	82.21
K-Means	Setting3	80.46	71.17	82.67	81.99
	Setting4	80.52	71.22	82.26	82.11
LDA	Setting3	82.97	71.32	85.14	83.65
	Setting4	82.41	71.06	85.91	82.78

The precision results in Table 3.4 demonstrate the effectiveness of Setting3 (using intra-clustering) and Setting4 compared with Setting2 with respect to medical history classification.

The highest precision of the intra-clustering algorithm was 85.91%, occurred when the NN classifier was used with unlabeled data. Comparing classifiers, the intra-clustering algorithm using NN has the highest improvement over the Raw representation. In this experiment, enriching the data set improved the results for the four classifiers, when the LDA method was used as the clustering algorithm. However, the intra-clustering using the K-Means method improves the results in AdaBoost, KNN, and the Setting3 of NN. The intra-clustering using K-Means with the SVM classifier with the TFiDF representation in Setting2 had slightly better performance than SVM in Setting3 and Setting4.

Table 3.5 shows the recall results of the proposed toolkit in different settings.

Table 3.5: Recall results of the learning classifiers.

Method	Experiment	AdaBoost	KNN	NN	SVM
Raw	Setting2	86.48	72.10	88.47	88.16
K-Means	Setting3	86.56	72.11	88.13	88.19
	Setting4	86.73	72.47	88.74	88.42
LDA	Setting3	88.49	73.16	89.72	89.82
	Setting4	88.69	72.84	92.71	91.49

These results show the improvement in the performance of the proposed approach. It is remarkable that the results for recall were higher than those for precision in all classifiers.

The results in Tables 3.4 and 3.5 show the superiority of using unlabeled data in the framework of the intra-clustering algorithm. The highest improvements were with the NN classifier in precision (0.77%) and recall (2.99%). Thus, when more data for clinical text classification are available, the proposed method has a higher chance of improving performance.

3.3.5 Comparison with LibShortText Software

LibShortText is a modification of the widely used *LibSVM* library, but for short text classification. The package includes the source code in *Python*2.6 and *C / C++*. The *LibShortText* software implements document search using the vector space model, then uses a bag-of-words model to generate future vectors. The *LibShortText* software uses the *LibSVM* with the advantage of not being required to tune the algorithm for the optimum kernel function and penalty factor.

Table 3.6 shows the comparison of classification performance using *LibShortText* and SALTClass for the UMCU text data in terms of macro- precision, recall and F1 score.

Table 3.6: Results for the SALTClass and the *LibShortText* software.

Software	Precision	Recall	F1
LibShortText	81.95	87.66	84.61
SALTClass	85.91	92.71	89.11

For given short texts, *LibShortText* follows the bag-of-words model to generate features, and preprocess short texts by tokenization, stemming, and stop-word removal. The library also allows users to choose between unigram and bigram features. However, *LibShortText* follows the routine pipeline of text classification

using the *LibSVM* library. On the contrary, SALTClass enriches the representation of short texts to improve the classification performance. As shown in Table 3.6, the SALTClass package gained better results compared with those on UMCU data from the *LibShortText* software.

3.4 Conclusions

EHRs contain a wealth of information in clinical text form. Thus, text mining techniques may be advantageously used to extract structured information. Medical data suffer from insufficient context, as they are represented in short texts. The SALTClass software was proposed to mitigate the classification error inherent in short texts by interpolating between observed and fitted counts obtained by clustering algorithm. SALTClass is a *Python* module for short text classification built under the MIT license. This module contains several functions for text preprocessing, cleaning, clustering, and classification. With the proposed intra-clustering algorithm, SALTClass can be fed with unlabeled data to incorporate the background knowledge for short texts. The SALTClass NLP toolkit enables users to apply various configuration combinations to their case study. To evaluate the effectiveness of SALTClass, we analyzed the classification of short cardiovascular notes collected in the UMCU hospital. It was demonstrated that using SALTClass can improve classification performance in terms of the precision, recall and F1 score.

Compliance with Ethical Standards

Funding: Folkert W. Asselbergs is supported by UCL Hospitals NIHR Biomedical Research Centre.

Conflict of Interest: The authors declare that they have no conflict of interest.

Chapter 4

Using Chest X-Ray Reports for Prediction of Recurrence of Major Cardiovascular Events

Bagheri, A., Groenhof, T. K. J., Asselbergs, F. W., Haitjema, S., Bots, M. L., Veldhuis, W. B., De Jong, P. A., & Oberski, D. L. (2020, submitted). Using Chest X-Ray Reports for Prediction of Recurrence of Major Cardiovascular Events in Cardiovascular Patients.

Bagheri, A., Groenhof, T. K. J., Veldhuis, W. B., De Jong, P. A., Asselbergs, F. W., & Oberski, D. L. (2020, in press). Multimodal Learning for Cardiovascular Risk Prediction using EHR Data. *In Proceedings of the 11th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB 2020).* `https://arxiv.org/abs/2008.11979`

Abstract

Electronic health records (EHRs) contain free-text information on symptoms, diagnosis, treatment, and prognosis of diseases. However, this potential goldmine of health information cannot be easily accessed and used unless proper text mining techniques are applied. The aim of this project was to develop and evaluate a text mining pipeline in a multimodal learning architecture to demonstrate the value of medical text classification in chest radiograph reports for cardiovascular risk prediction. We sought

Author contributions: AB and TKJG designed the study. AB developed the statistical methods and source code. AB and TKJG analyzed the data and experiments. AB and TKJG wrote the paper. FWA, SH, MLB, WBV, PAdJ and DLO provided feedback on written work.

to assess the integration of various text representation approaches and clinical structured data with state-of-the-art deep learning methods in the process of medical text mining.

We used EHR data of patients included in the Second Manifestations of ARTerial disease (SMART) study. We propose a deep learning-based multimodal architecture for our text mining pipeline that integrates neural text representation with preprocessed clinical predictors for the prediction of recurrence of major cardiovascular events in cardiovascular patients. Text preprocessing, including cleaning and stemming, was first applied to filter out the unwanted texts from x-ray radiology reports. Thereafter, text representation methods were used to numerically represent unstructured radiology reports with vectors. Subsequently, these text representation methods were added to prediction models to assess their clinical relevance. In this step, we applied logistic regression, support vector machine (SVM), multi-layer perceptron neural network, convolutional neural network, long short-term-memory (LSTM), and bidirectional LSTM deep neural network (BiLSTM).

We performed various experiments to evaluate the added value of text in the prediction of major cardiovascular events. The two main scenarios were the integration of radiology reports (1) with classical clinical predictors (2) and with only age and sex in the case of unavailable clinical predictors. In total, data of 5603 patients were used with 5-fold cross validation to train the models. In the first scenario, the multimodal BiLSTM (MI-BiLSTM) model achieved an area under curve (AUC) of 84.7%, misclassification rate of 14.3%, and F1 score of 83.8%. In this scenario, the SVM model, trained on clinical variables and bag-of-words representation, achieved the lowest misclassification rate of 12.2%. In the case of unavailable clinical predictors, the MI-BiLSTM model trained on radiology reports and demographic (age and sex) variables reached an AUC, F1 score, and misclassification rate of 74.5%, 70.8%, and 20.4%, respectively.

Using the case study of routine care chest x-ray radiology reports, we demonstrated the clinical relevance of integrating text features and classical predictors in our text mining pipeline for cardiovascular risk prediction. The MI-BiLSTM model with word embedding representation appeared to have a desirable performance when trained on text data integrated with the clinical variables from the SMART study. Our results mined from chest x-ray reports showed that models using text data in addition to laboratory values outperform those using only known clinical predictors.

4.1 Introduction

Electronic health records (EHRs) data have become increasingly available to researchers as more hospitals, clinics and practices have adopted data digitization. EHRs store data in different modalities, such as structured data (e.g. demographic values, laboratory results, medications) and unstructured texts (e.g. referral letters, clinical notes, discharge summaries, radiology reports). This digitization creates an opportunity to mine the health records to increase the quality of care and clinical outcomes. Yet clinicians have limited time to process all the available data and detect patterns across similar medical records. Deep learning and machine learning, on the other hand, are suitable for discovering useful patterns from vast amount of data.

Unstructured texts contained within the EHRs are recognized as a rich but not easily accessible and usable source of medical information (Sheikhalishahi et al., 2019; Drozdov et al., 2020; Zech et al., 2018; Z. Wang et al., 2012; Pérez, Pérez, Casillas, & Gojenola, 2018). Recent studies have attempted to derive information from unstructured medical texts to classify disease codes (Du et al., 2019; Bagheri, Sammani, Van der Heijden, Asselbergs, & Oberski, 2020a; Blanco, Perez-de Viñaspre, Pérez, & Casillas, 2020; J. Huang, Osorio, & Sy, 2019), detect patient's disease history (Bagheri, Sammani, Van der Heijden, Asselbergs, & Oberski, 2020b; X. Wu et al., 2020), and predict hospital readmission or clinical outcomes (Alex et al., 2019; K. Huang et al., 2019; Jagannatha & Yu, 2016). X-ray radiology reports are example of such unstructured data describing radiologist's observations on patient's medical conditions associated to medical images. The majority of previous decision support systems for radiology reports are developed using rule-based approaches applied on unstructured and semi-structured texts (Gong et al., 2008; M. C. Chen et al., 2018; Taira et al., 2001; Y. Wang et al., 2018). However, these methods are often impractical because they do not generalize to new data and often are not applicable for big data analysis (Pons et al., 2016).

Recent studies have shown promising results using free-text radiology reports and deep learning models to predict clinical outcomes (M. C. Chen et al., 2018; Monshi, Poon, & Chung, 2020; Laserson et al., 2018; Smit et al., 2020; Wood et al., 2020; Shin, Chokshi, Lee, & Choi, 2017). Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are two common deep learning techniques that have been effective in text mining and natural language processing (NLP), as well as EHR applications (Du et al., 2019; Jagannatha & Yu, 2016; Monshi et al., 2020; Banerjee et al., 2019; Moradi, Dorffner, & Samwald, 2020). Deep learning-based modeling of radiology reports has been proposed to supersede the simple grammatical patterns and hand-crafted regular expressions of the traditional clinical rules-based software, such as PEFinder (Chapman, Lee, Kang, & Chapman, 2011), MedLEE (Hripcsak, Austin, Alderson, & Friedman, 2002; C. Friedman,

Alderson, Austin, Cimino, & Johnson, 1994), and CTakes (Savova et al., 2010). While these neural networks models gained tremendous momentum in knowledge discovery from EHR texts, there are very seldom studies that used of both free-texts and structured information in EHRs for clinical prediction and classification (Scheurwegs et al., 2016; Liu et al., 2018; Xu et al., 2018; Jin et al., 2018).

In this paper we leveraged structured features in EHR data to combine with free-text radiology reports to uncover patterns to improve cardiovascular risk prediction. Free-text within EHRs might contain additional information for clinical prediction modeling, either as an added variable to improve prediction performance compared to current models or as an auxiliary variable to increase the flexibility of prediction in the case of inaccessible clinical data.

The contributions of this study are two-fold. The first contribution is to develop and evaluate a text mining pipeline for capturing additional information from text. The second is the use of chest x-ray reports from routine care as free-text in combination with the laboratory values, collected in the Second Manifestations of ARTerial disease (SMART) study (Simons, Algra, Van de Laak, Grobbee, & Van der Graaf, 1999), in a multimodal architecture to predict the recurrence of cardiovascular events in cardiovascular patients.

4.2 Materials and Methods

In this section, we describe the case study, data ethics and privacy, and the details of our proposed text mining pipeline.

4.2.1 Case Study

4.2.1.1 Patient Population

The patients included in this study were originally included in the SMART study. The design of the SMART study is published elsewhere (Simons et al., 1999). In short, the SMART study is an ongoing single-center prospective cohort study designed to establish the presence of additional arterial disease and risk factors for atherosclerosis in patients with vascular disease or a vascular risk factor. Patients visiting the University Medical Center (UMC) Utrecht for evaluation of any atherosclerotic cardiovascular condition are eligible for inclusion in SMART. The inclusion criteria are presentation with an atherosclerotic cardiovascular condition and age > 18 years. Exclusion criteria are life expectancy < 3 months, unstable vascular disease, and insufficient fluency in the Dutch language. A total of 5603 SMART patients were included in this analysis. The characteristics of the patients are listed in Table 4.1.

Table 4.1: Characteristics of the patients.

Characteristic	Total (n = 5603)
Age, years, mean (sd)	56.2 (12.5)
Female sex, n (%)	1926 (34.4)
Current smoker, n (%)	1549 (27.6)
History of CVD[a]	
CHD[b], n (%)	2166 (38.7)
Stroke, n (%)	1076 (19.2)
PAD[c], n (%)	631 (11.3)
AAA[d], n (%)	306 (5.5)
Years since first diagnosis of CVD, median (IQR)	0 (0-4)
Risk factors for CVD	
Diabetes Mellitus, n (%)	1047 (18.7)
Hypertension, n (%)	2353 (42.0)
Dyslipidemia, n (%)	432 (7.7)
BMI[e], kg/m2 (mean (sd))	26.8 (4.3)
SBP[f], mmHg (mean (sd))	140 (21)
DBP[g], mmHg (mean (sd))	83 (13)
Total cholesterol, mmol/L (mean (sd))	5.14 (1.38)
LDL[h]-cholesterol, mmol/L (mean (sd))	3.1 (1.16)
HDL[i]-cholesterol, mmol/L (mean (sd)	1.27 (0.38)
Triglycerides, mmol/L (median (IQR))	1.7 (1.2-2.5)
MDRD[j], ml/min/1.73m2 (median (IQR))	80 (68-91)
HbA1c[k], mmol/mol (median (IQR))	5.7 (5.4-6.1)
Glucose, mmol/L (median (IQR))	5.7 (2.6-6.4)
Hemoglobin, mmol/L (mean (sd))	6.0 (2.04)
Creatinine, μmol/L (median (IQR))	84 (73-97)
CRP[l], mg/L (median (IQR))	1.95 (0.90-4.20)
TSH[m], mU/l (mean (sd))	0.9 (0.09)
MACE[n] during followup, n (%)	1385 (24.7)

[a]CVD: Cardiovascular disease [b]CHD: Coronary heart disease [c]PAD: Peripheral arterial disease [d]AAA: Abdominal aortic aneurysm [e]BMI: Body mass index [f]SBP: Systolic blood pressure [g]DBP: Diastolic blood pressure [h]LDL: Low-density lipoprotein [i]HDL: high-density lipoprotein [j]MDRD: Modification of diet in renal disease [k]HbA1c: Hemoglobin A1c [l]CRP: C-reactive protein [m]TSH: Thyroid stimulating hormone [n]MACE: Major cardiovascular events

4.2.1.2 Clinical Variables

Variables that are predictors in the SMART study (Simons et al., 1999) (age; sex; smoker; systolic blood pressure; diabetes; HDL cholesterol; total cholesterol; renal function according to the MDRD formula; history of cardiovascular disease stratified for stroke, peripheral artery disease, abdominal aortic aneurysm, and coronary heart disease; and years since diagnosis of first cardiovascular disease) were used for prediction modeling for all patients.

4.2.1.3 Chest X-Ray Reports

Free-text reports from chest x-rays that were taken in SMART patients, that were made in routine care, were extracted from their EHR and included in this analysis.

4.2.1.4 Ethics and Privacy

Informed consent was obtained through established procedures. The SMART study was approved by the Medical Ethical Committee of the UMC Utrecht. All data are handled according to local data protection guidelines and privacy regulations.

4.2.2 Text Mining Pipeline

Figure 4.1 illustrates the text mining pipeline for the prediction task. The goal is to forecast the major cardiovascular events (MACE) during follow-up as the outcome prior to clinical variables and chest x-ray reports.

4.2.2.1 Preprocessing

Clinical variables were pre-processed by missing value imputation and a normalization step. Missingness of data was solved using the $MICE$ package (Buuren & Groothuis-Oudshoorn, 2010) with one imputation for each missing value. As an additional normalization step, the clinical variables were re-scaled to homogenize their levels of variance.

In preprocessing the radiology reports, the following steps were performed to improve the quality of text data for the subsequent steps: (1) All characters were transformed into lowercase. (2) We removed numbers and some meaningless punctuation marks, such as semicolons and colons. (3) Stop words were then removed. Dutch stop words used in this study are shown in the Appendix. (4) We then applied Porter's stemming algorithm (Kraaij & Pohlmann, 1994; Porter, 2001) to texts. Figure 4.2 shows the 20 most frequent words before and after the preprocessing step for the x-ray radiology report in the SMART study. "klinisch (clinical)" and "xthorax (chest x-ray)" appeared in all reports as they denote the indication

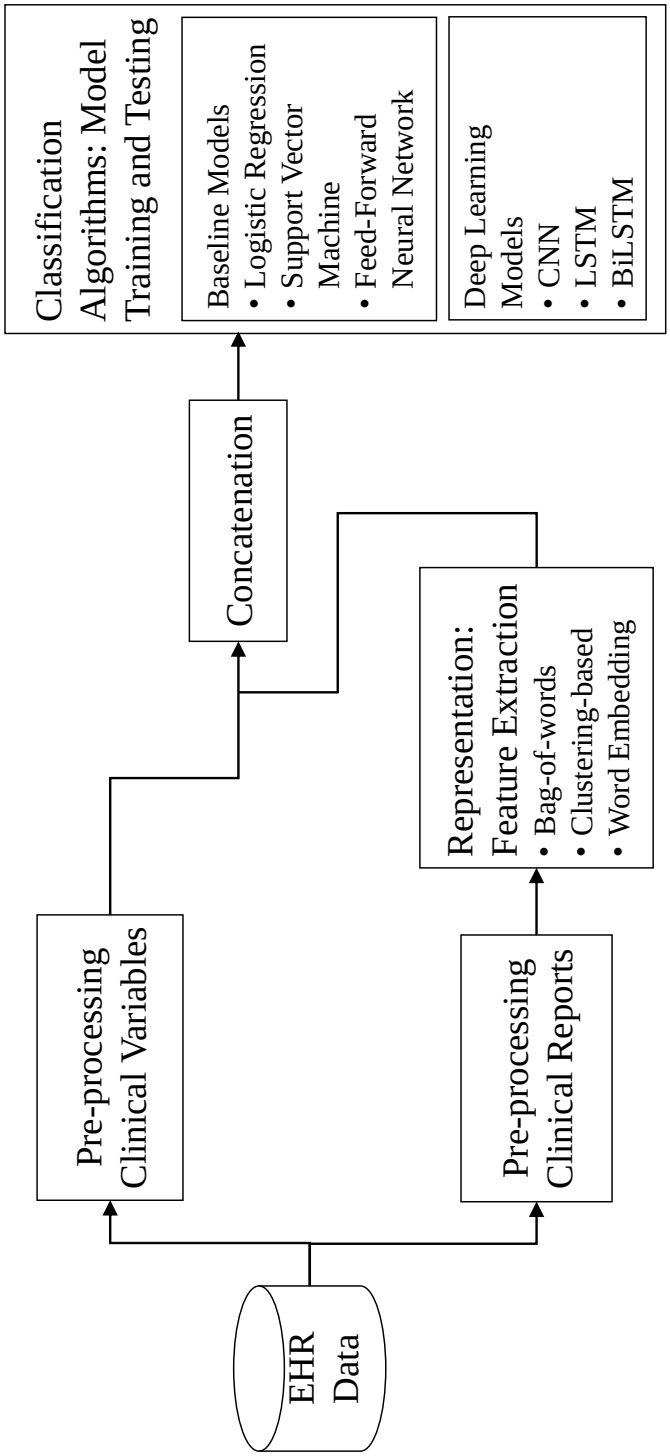
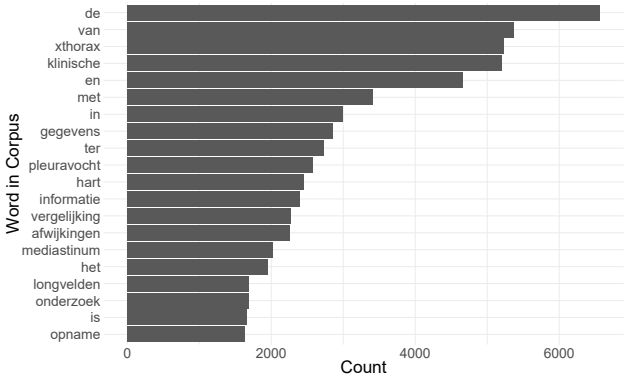
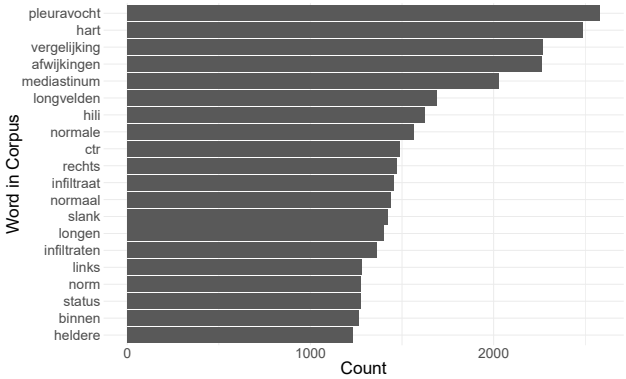


Figure 4.1: Methodology text mining pipeline overview.



(a) Initial top frequent words



(b) Top frequent words after preprocessing

Figure 4.2: Most frequent words in the x-ray radiology reports in the SMART study.

of the test and type of x-ray; therefore, we removed them as non-informative stop words. Other words were merged into their stem words.

4.2.2.2 Representation and feature extraction

Text representation includes dimensions in which text is represented in a vector space model. We explored three text representation techniques used in the text mining pipeline:

- Bag-of-words (BOW)
- Clustering-based representation
- Word embedding

We used three different techniques: an interpretable method, a method with a low-dimensional output, and a less interpretable and more semantic-based technique to be able to assess their differences in performance in mining additional information for clinical prediction modeling.

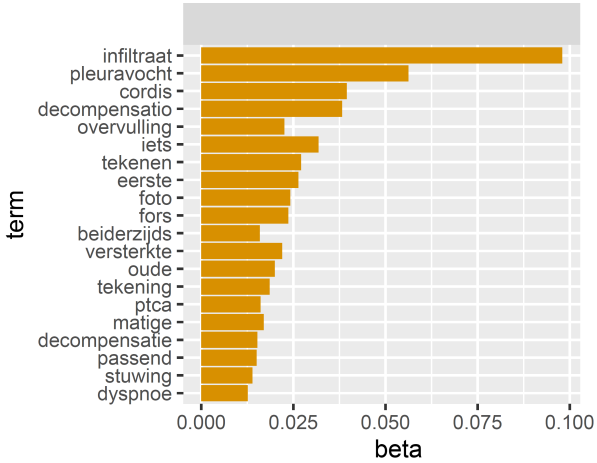
Bag-of-Words The BOW representation is the most commonly used representation for text mining applications (Bagheri et al., 2020b). Words in the reports were converted into a sparse multidimensional representation, which was leveraged for further classification and clustering purposes. Representation of text includes frequencies of words per patient's text report. This is a method that is relatively easy to understand and interpret for clinicians.

Clustering-based Representation We applied latent Dirichlet allocation (LDA) (Blei et al., 2003) to further cluster the BOW representation of patient radiology reports. LDA is a topic modeling approach that groups a collection of documents to obtain the probabilities of the distributions of document–topic and topic–word in the data set. This method has the advantage of using an interpretable lower dimensional representation of text and the disadvantage of lacking the capacity of methods that use all features of unstructured medical notes. We ran the experiments fitting the LDA topic model with Gibbs sampling (Bagheri et al., 2020b; Blei et al., 2003) using 10 topics. Figure 4.3 shows two topics of the output of LDA applied to the x-ray radiology reports in the SMART study. Potential clinical scenarios that fit these topics are a) possible cardiac decompensation, and b) possible pneumonia.

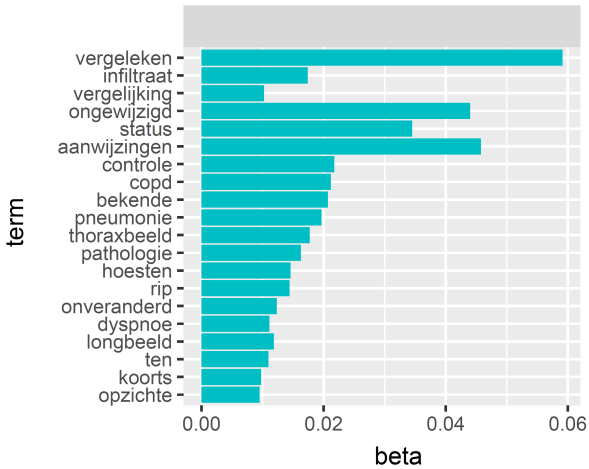
Word Embedding Neural network-based word embedding incorporates not only the contexts of a word but also the semantic relation with other words (Sheikhalishahi et al., 2019; Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013). We used a window of five words for the context words that were used in this representation technique. Subsequently, word vectors were aggregated for each patient report.

4.2.2.3 Classification Algorithms

In the text mining pipeline, independent variables are the clinical variables and features extracted from radiology reports, though these features differ per text representation approach. MACE as defined by the SMART study was used as an



(a) Possible cardiac decompensation



(b) Possible pneumonia

Figure 4.3: LDA clustering: The y-axis shows the top words in the selected cluster (topic). The x-axis shows the probability of the word in the topic.

outcome variable. We made a total of six different algorithms to be able to study both baseline and easier to interpret and state-of-the-art yet more opaque machine learning methods and their additional value for clinical risk prediction.

Baseline Models Using traditional machine learning classifiers, we applied an LR model and a support vector machine (SVM) algorithm to data from the SMART case study. If the interpretation of a model is of primary interest, LR parameters can easily be interpreted in terms of the log odds. SVM on the other hand is a supervised learning technique that produces nonlinear boundaries by constructing a linear boundary in a large, transformed version of the feature space, and it scales relatively well to high dimensional data, such as unstructured texts (Tong & Koller, 2001).

Deep Learning Models We studied using state-of-the-art deep learning methods: a CNN, a long short-term memory (LSTM) RNN, and a bidirectional LSTM (BiLSTM) (Du et al., 2019; Jagannatha & Yu, 2016; Monshi et al., 2020; Banerjee et al., 2019). We also employed a feed-forward multi-layer perceptron neural network for the case of no text presented to the model. However, multilayer perceptron is not well adapted to textual data (Sheikhalishahi et al., 2019; Bagheri et al., 2020a; Banerjee et al., 2019). This is because it is defined for vectors as input data; hence, to apply it to texts, we must transform the texts into vectors. CNN, LSTM, and BiLSTM are deep learning architectures that have removed the manual extraction of features from text data.

Multimodal Neural Network Figure 4.4 illustrates the proposed deep learning-based architecture for the text mining pipeline. In this architecture, we propose a multimodal learning model using a BiLSTM deep neural network. The multimodal neural network architecture consists of an embedding layer, a BiLSTM layer, dropout, a concatenation layer and dense layers.

Embedding Layer To extract the semantic information of radiology reports, each text is firstly represented as a sequence of word embeddings. Word embedding is an improvement over the bag-of-words models where large sparse vectors were used to represent each word. On the contrary, in an embedding, words are represented by dense vectors where a vector represents the projection of the word into a continuous vector space (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013). Denote s as an X-ray report with m words and each word is mapping to a vector, then we have:

$$s = [\vec{e}_1, \vec{e}_2, ..., \vec{e}_m] \tag{4.1}$$

where vector $\vec{e}_i$ represents the vector of i-th word with a dimension of d. The

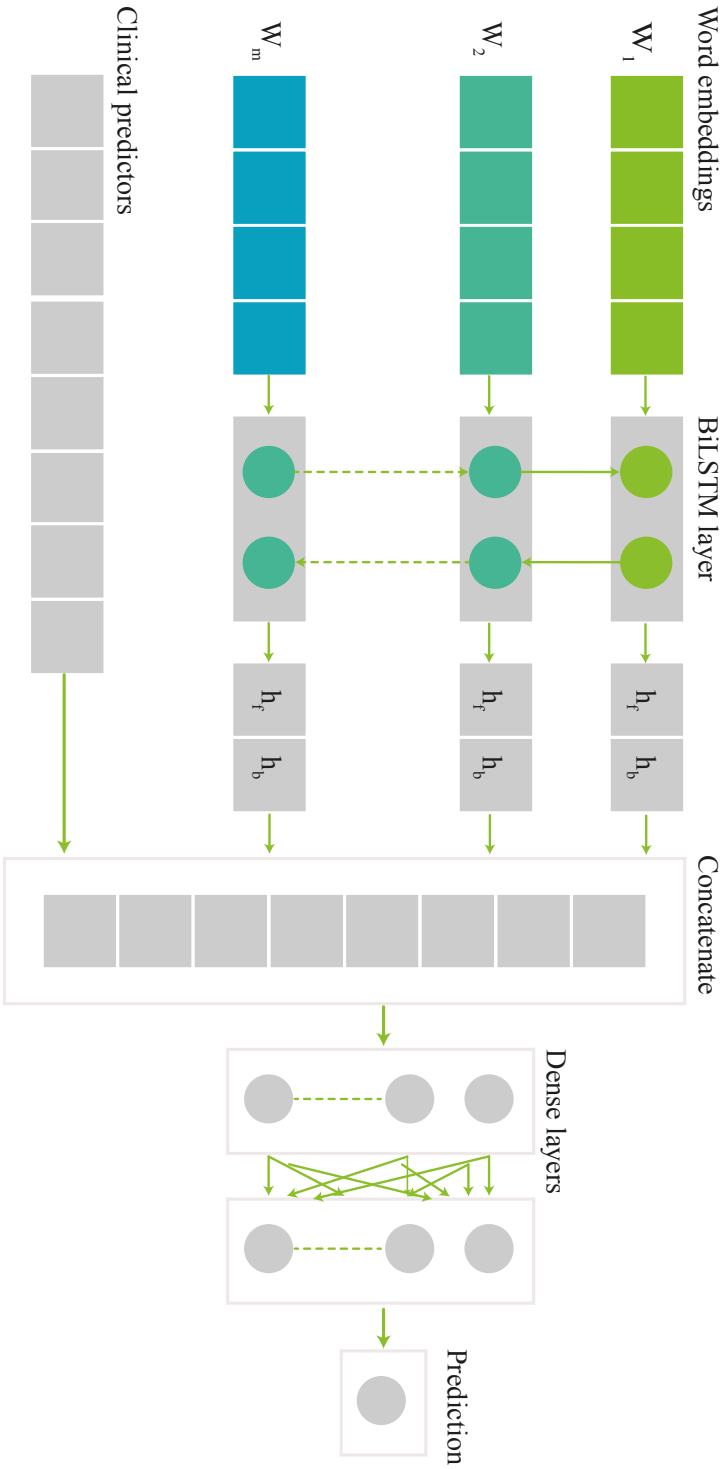


Figure 4.4: Proposed multimodal learning architecture with a deep learning model.

vectors of word embeddings are concatenated together to maintain the order of words in a patient report.

Bidirectional-LSTM Layer After the embedding layer, the sequence of word vectors is fed into a bidirectional LSTM layer to achieve another representation of radiology reports. Interest in incorporating a BiLSTM layer into the architecture of our model arises from their ability to learn long-term dependencies and contextual features from both past and future states (Schuster & Paliwal, 1997). The BiLSTM layer calculates two parallel LSTM layers, a forward hidden layer and a backward hidden layer, to generate an output sequence y as illustrated:

$$h_{f_t} = \sigma(W_{xh_f} x_t + W_{h_f h_f} h_{f_{t-1}} + b_{h_f}) \tag{4.2}$$

$$h_{b_t} = \sigma(W_{xh_b} x_t + W_{h_b h_b} h_{b_{t-1}} + b_{h_b}) \tag{4.3}$$

$$y_t = W_{h_f y} h_{f_t} + W_{h_b y} h_{b_t} + b_y \tag{4.4}$$

Here σ is the sigmoid activation function, x_t is a d-dimensional input vector at time step t, W are the weight matrices, b are bias vectors, and h_f, h_b are the output of the LSTM forward and backward layers, respectively.

The multimodal BiLSTM integrates the neural text representation with clinical predictors and feeds them into a fully connected neural network. We used a BiLSTM network to connect both previous and future information to the present information in text reports. This was made possible by having two propagating networks in opposite directions: one network running from the beginning of the text to the end and the other in the opposite direction. These forward and backward networks memorize information about the report from both directions. Thus, the context window around each word consists of both information prior to and after the current word. In this way, BiLSTM can model the entire sequence of words in a radiology report to capture dependencies between the feature space and the relationship with the outcome variable.

Other Deep Neural Networks When applying a CNN model to our architecture, we used a convolution layer with a max pooling layer instead of the BiLSTM layer in the architecture in Figure 4.4. For employing an LSTM model, only the left to right direction in text is monitored inside the hidden RNN layer.

4.2.3 Evaluation Measures

To evaluate the classification performance of our text mining pipeline, we used five available metrics: area under the curve (AUC), misclassification rate, precision (positive predictive value), recall (sensitivity), and F1 score. AUC is the area

under the receiver operating characteristic curve, which is created by plotting the true positive rate against the false positive rate. Misclassification rate is the proportion of incorrectly classified instances made by a model. Precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. The F1 score can be interpreted as a weighted average of the precision and recall. The relative contributions of precision and recall to the F1 score are equal. The formulae of precision, recall, and the F1 score are defined in Equations 4.5, 4.6, and 4.7:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4.5}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{4.6}$$

$$\text{F1 score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.7}$$

4.3 Results

Our pipeline was implemented in *Python* and *R* using various text mining, NLP, and machine learning packages. The multimodal learning architecture was implemented on *Keras* with a *TensorFlow* backend[1]. The source code is publicly available at GitHub[2]. We performed 5-fold cross validation for all experimental analyses. We used the embeddings with a vector size equal to 500 and a window size equal to 5. In addition, we set the number of filters in the CNN to 128 and the filter size to 5. The hidden dense layers contained 64 units and used the *ReLU* activation function, and the output layer used a *sigmoid* activation function. We set the same number of hidden units in the LSTM layers at 100. Dropout and recurrent dropout were added both at 0.2 to avoid overfitting (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). We set the batch size and number of epochs to 64 and 20, respectively. These hyperparameters were tuned based on the validation set.

To assess the added value of text for the prediction of MACE, we compared various scenarios of clinical variables and text reports in the proposed text mining pipeline:

1. Prediction using only radiology reports (models starting with T)
2. Prediction using only clinical variables (models starting with V)

[1] `https://keras.io`
[2] `https://github.com/bagheria/CardioRisk-TextMining`

3. Prediction using the integration of clinical variables and radiology reports (models starting with VB, VC, and MI)
4. Prediction using only sex and age variables (models starting with D)
5. Prediction using the integration of sex and age variables and radiology reports (models starting with DB, DC, and D-MI)

Table 4.2 lists the experimental results for AUC and misclassification rate for the first three scenarios. In these experiments, we evaluated different models using only clinical variables, only radiology reports, and their integration.

Table 4.2: Performance comparison of different experimental scenarios using AUC and misclassification rate.

Classifier	AUC	Misclassification rate
V-LR	0.799	0.195
V-SVM	0.648	0.196
V-NN	0.651	0.201
T-LR	0.512	0.247
T-SVM	0.625	0.186
T-BiLSTM	0.570	0.300
VB-LR	0.808	0.193
VB-SVM	0.784	0.122
VC-LR	0.809	0.194
VC-SVM	0.655	0.197
MI-LR	0.811	0.203
MI-SVM	0.694	0.237
MI-CNN	0.730	0.214
MI-LSTM	0.794	0.176
MI-BiLSTM	0.847	0.143

V-LR, V-SVM, and V-NN are the models trained on only clinical variables. The features in these models included the SMART variables as independent variables and MACE during follow-up as the outcome in prediction models. T-SVM, T-LR, and T-BiLSTM are the models with only text reports as their predictors. T-SVM was trained on the BOW representation of the reports. T-LR used the clustering-based representation. In this scenario, we reported each model's best result among representation methods. T-SVM achieved the highest performance in this scenario with an AUC of 62.5% and misclassification rate of 18.6%. VB and VC are the models trained on clinical variables combined with the BOW and clustering-based representations, respectively. VC-SVM gained the lowest AUC of 65.5%, while the VB-SVM model obtained the lowest misclassification rate of 12.2%.

MI represents the models that used of the proposed multimodal learning architecture with the neural word embedding representation. In this scenario, MI-

BiLSTM, MI-LSTM, and MI-LR achieved promising results. MI-BiLSTM obtained the highest AUC of 84.7% and the lowest misclassification rate of 14.3% in this case. MI-LR still has the second ranking AUC at 81.1%.

Precision, recall and F1 score evaluation measures are recommended for imbalanced data, where the AUC and misclassification rate may provide an optimistic view of the performance (Branco, Torgo, & Ribeiro, 2015). Figure 4.5 [3] shows the performance of the models using precision, recall, and F1 score metrics. The deep learning models achieved better performance compared to other models in different scenarios. The MI-BiLSTM model achieved the highest performance in terms of all evaluation measures. The F1 score was 83.8%. MI-LSTM and MI-CNN obtained F1 score performances of 78.9% and 74.7%, respectively. These results are evidence of the performance of text mining techniques with multimodal learning architecture in extracting knowledge from radiology reports and combining them with classical clinical predictors. It is notable that the multi-layer perceptron neural network achieved promising results when trained only on clinical variables. This model obtained a precision of 75.1%, recall of 79.4, and F1 score of 77.2%. This shows the efficiency of the neural network model and the relatedness of the laboratory results in predicting cardiovascular risk.

To assess the value of text as additional variables if clinical predictors are not available, we again compared the above-mentioned scenarios but with only sex and age as clinical variables. Table 4.3 lists the results of this evaluation of the text mining pipeline. We named the models in this scenario D-models to show that they have been trained on demographic (age and sex) features.

The D-MI-BiLSTM model gained the highest AUC of 74.5%. D-MI-BiLSTM was trained using the multimodal architecture, meaning that it used the neural word embedding representation and BiLSTM hidden layer output to concatenate the radiology reports with age and sex. DB-SVM gained the lowest misclassification rate of 16.3%. This model was trained on the combination of the BOW representation and the age and sex variables.

In Figure 4.6, the results of precision, recall, and F1 score are compared for the scenarios when clinical predictors are not available. This setting also confirms that text mining-based models achieved better performance when predicting the MACE variable. The D-MI-BiLSTM, D-MI-LSTM, and D-MI-CNN models gained F1 scores of 70.8%, 67%, and 64.3%, respectively. The LR model with only age and sex only reached to 44.2%, 49.4%, and 46.66%, respectively.

4.4 Discussion

This study aimed to develop and evaluate a text mining pipeline integrating clinical and text variables applied to cardiovascular risk prediction. Our research: (1)

[3] Firatheme version 0.2.1 (van Kesteren, 2020)

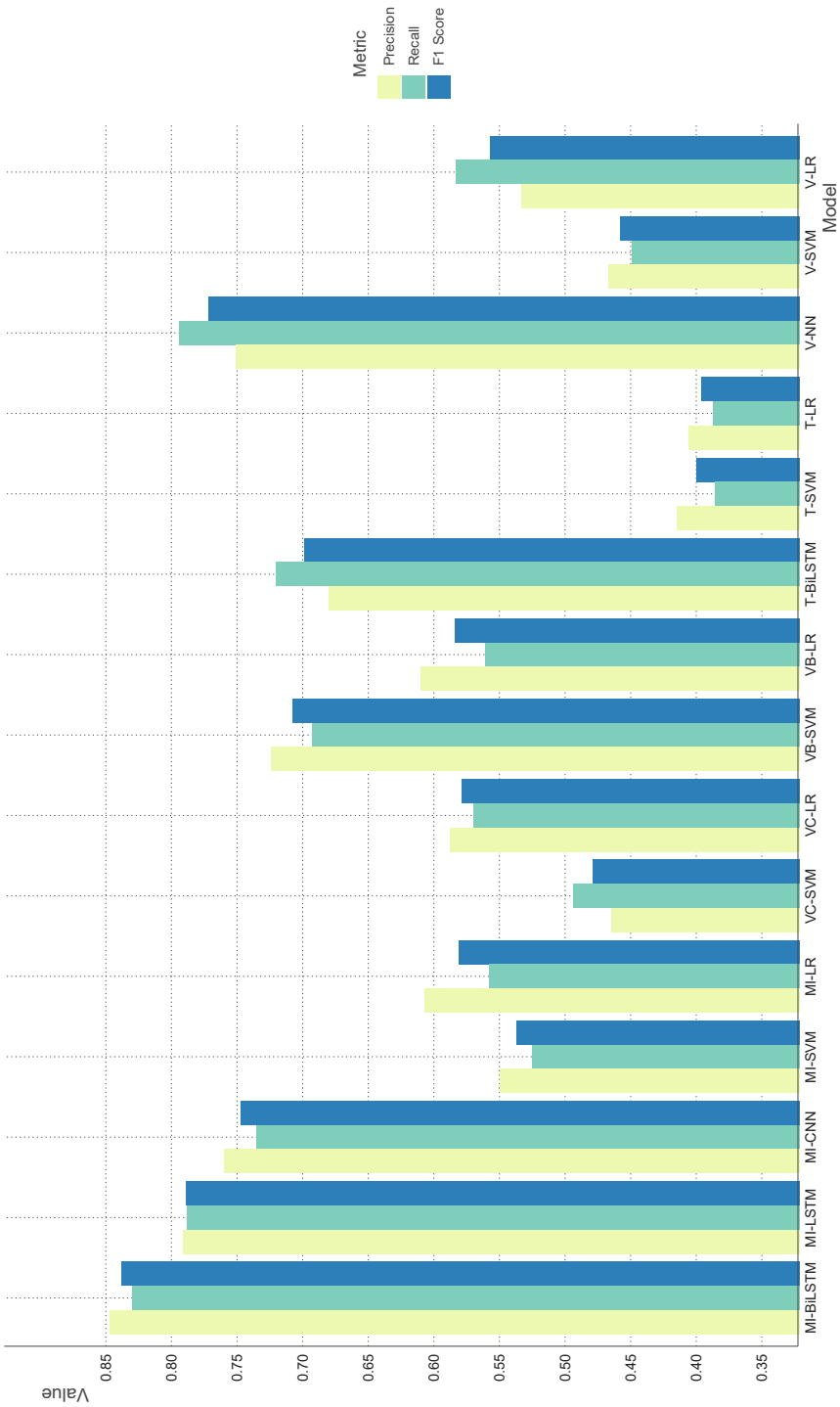


Figure 4.5: Comparison of precision, recall, and F1 score for experimental scenarios.

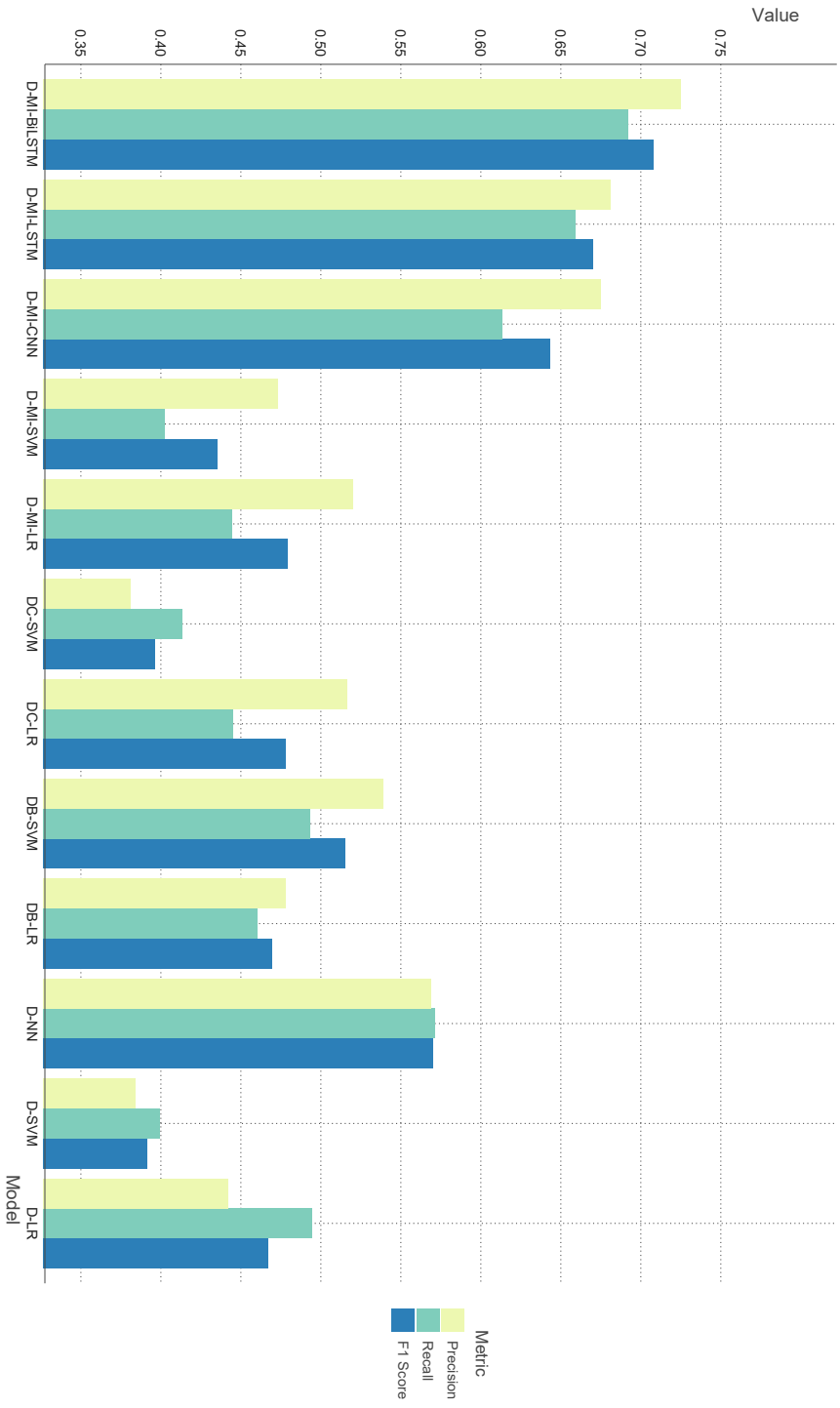


Figure 4.6: Comparison of precision, recall, and F1 score for experimental scenarios when clinical predictors are not available.

Table 4.3: Performance comparison of different experimental scenarios using AUC and misclassification rate when clinical predictors are not available.

Classifier	AUC	Misclassification rate
D-LR[a]	0.685	0.242
D-SVM	0.572	0.246
D-NN	0.567	0.214
DB-LR[b]	0.703	0.247
DB-SVM	0.674	0.163
DC-LR[c]	0.705	0.239
DC-SVM	0.534	0.247
D-MI-LR[d]	0.708	0.235
D-MI-SVM	0.568	0.247
D-MI-CNN	0.667	0.228
D-MI-LSTM	0.708	0.209
D-MI-BiLSTM	0.745	0.204

[a]LR trained on demographic variables. [b]LR trained on demographic variables and BOW representation. [c]LR trained on demographic variables and clustering-based representation. [d]Multimodal learning LR trained on demographic variables and word embeddings.

integrates EHR structured laboratory results and unstructured radiology text in a text mining pipeline to make an accurate decision for classifying cardiovascular events; (2) uses routine care data including x-ray reports in Dutch for cardiovascular risk prediction, and (3) incorporates free-text reports as auxiliary variables when classical predictors are not available. In our experiments for the SMART case study, we found that neural text representation and prediction modeling significantly add to baseline models with classical clinical predictors to predict MACE. In the case of unavailable clinical predictors, the proposed MI-BiLSTM model with just age, sex, and word embedding attains a similar discriminative performance to that of the models trained on the classical SMART variables. Deep learning methods are increasingly being adopted in the medical field. For example, in radiology, deep learning has shown remarkable results in image analysis (van Amsterdam, Verhoeff, de Jong, Leiner, & Eijkemans, 2019), and in intensive care, RNNs have been used to determine variables that are proxies for clinician decision-making (Lin, Zhou, Faghri, Shaw, & Campbell, 2019). The application of text mining and NLP in predictive setting is not new; unlocking the full potential of EHR data is contingent on the development of text mining pipelines to automatically transform free-text into structured clinical data that can guide clinical decisions (Sheikhalishahi et al., 2019; Drozdov et al., 2020; Z. Wang et al., 2012). Yet, text as auxiliary variables to classical clinical variables has only been considered in a few studies (Scheurwegs et al., 2016; Liu et al., 2018; Xu et al., 2018; W. Chen et al., 2020; Suresh et al., 2017). One study (Suresh et al., 2017) predicted several clinical interventions combining structured data and clinical notes. Each clinical

narrative note was transformed to a 50-dimensional vector of topic proportions for each note using an LDA algorithm. This resulted in a lower dimensional representation of text, losing the depth of information in unstructured text. Another study (Liu et al., 2018) extracted structured information from clinical notes using regular expression and a heuristic rule-based tool. Again, this is a method that uses a general framework for predicting the onset of diseases, combining both free-text medical notes and structured information. The mined text was then used to predict congestive heart failure, kidney failure, and stroke via deep learning models, achieving good performance in disease prediction. Lastly, one study (Xu et al., 2018) combined unstructured text, semi-structured text, and structured data in machine learning models. Separate models were developed to handle data from different modalities to create an ensemble model that predicts diagnostic codes of international classification of diseases (ICD-10). Hence in this study, we combine all advantages of prior research by developing a machine learning-based modeling of radiological language, to integrate clinical variables and textual features, to supersede traditional algorithms using only clinical variables. In this paper, we explained how we used text preprocessing techniques and applied text representation methods to chest x-ray reports. These representations were then used as auxiliary variables to the clinical variables from the SMART study to predict MACE using six different classification techniques.

There are strengths and limitations to our case study. Because patients must have had an indication for a routine care chest x-ray, there was a selection in the case study. However, this does not in fact mean there is selection bias; it merely restricts generalizability of the clinical prediction model to cardiovascular patients without an x-ray report available. Pragmatically, we hypothesize that using information that is available – including bodies of text, such as this chest x-ray report – for predictions rather than a strict set of predictors, will make predictions more flexible and more tailored to individuals. The use of advanced techniques, such as text mining, in clinical practice requires support for implementation. Implementation includes the application of the mining pipeline and integration in the care process using technologies, such as computerized decision support (CDSS). CDSS allows technical results from algorithms, such as text mining, to be translated to practical suggestions for clinical practice. To help clinicians interpret results that come from text mining, collaborations between technical text mining experts (bio-statisticians, mathematicians, data scientists, and software engineers) and practical experts (clinicians) are needed to safeguard technical quality and medical relevance. Future studies will focus on two points. First, our multimodal learning architecture will be validated for other similar scenarios, such as adverse event monitoring, hospital readmission, or disease classification, in which both EHR structured variables and free-text reports would contribute to the judgement of final outcomes. Secondly, we will expand our pipeline to a model to use the available clinical dictionaries with machine learning and deep learning models.

The publicly available source code of our model[4] can be used to evaluate performance on clinically-relevant classification tasks based on clinical notes and EHR variables.

4.5 Conclusions

Medical free-text potentially contains valuable information for clinical decision making. Text mining methods are the key to successful extraction of clinically important findings from these free-text reports. Medical text mining is a step-by-step process that requires tailoring to the aim of the project and the context of reports. Text mining potentially opens the door to valuable information captured in free-text medical data. We believe that such models are useful in reducing work overload for clinicians by providing needed clinical decision support.

Compliance with Ethical Standards

Funding: This work has received support from the EU/EFPIA Innovative Medicines Initiative 2 Joint Undertaking BigData@Heart grant n° 116074. The author Folkert W. Asselbergs is supported by UCL Hospitals NIHR Biomedical Research Centre. The Department of Radiology partly receives research support from Philips Healthcare.

Conflict of Interest: The authors declare that they have no conflict of interest.

Acknowledgments: We thank Erik-Jan van Kesteren for his helpful comments on an earlier version of this manuscript.

[4] `https://github.com/bagheria/CardioRisk-TextMining`

Appendix 4.A Dutch Stop Words

de	informatie	je	al	na
worden	tegen	en	eerdere	mij
waren	reeds	zelf	over	van
klinisch	uit	doen	wil	ons
klinische	ik	er	der	toen
kon	kunnen	gegeven	gegevens	maar
daar	moet	uw	ook	ja
dat	om	haar	ben	ter
bij	ge	die	hem	naar
kan	geweest	zich	nu	in
dan	heb	hun	andere	te
had	aan	zou	hoe	dus
iemand	voor	als	een	of
heeft	onder	informatie	hier	eens
hij	wat	hebben	omdat	tot
men	u	het	mijn	deze
thorax	xthorax	zijn	doch	is
dit	want	wie	conclusie	met
me	was	zo	nog	werd
onderzoek	ze	zij	op	door
zal	altijd	opname	wordt	eerder
x /X				

Chapter 5

Automatic ICD-10 Classification of Diseases from Dutch Discharge Letters

Bagheri, A., Sammani, A., Van der Heijden, P. G. M., Asselbergs, F. W., & Oberski, D. L. (2020). Automatic ICD-10 Classification of Diseases from Dutch Discharge Letters. *In Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2020.* 13, 281–289. doi.org/10.5220/0009372602810289

Abstract

The international classification of diseases (ICD) is a widely used tool to describe patient diagnoses. At University Medical Center Utrecht (UMCU), for example, trained medical coders translate information from hospital discharge letters into ICD-10 codes for research and national disease epidemiology statistics, at considerable cost. To mitigate these costs, automatic ICD coding from discharge letters would be useful. However, this task has proven challenging in practice: it is a multi-label task with a large number of very sparse categories, presented in a hierarchical structure. Moreover, existing ICD systems have been benchmarked only on relatively easier versions of this task, such as single-label performance and performance on the higher "chapter" level of the ICD hierarchy, which

Author contributions: AB and DLO designed the study. AB developed the statistical methods and source code. AB and AS analyzed the data and experiments. AB wrote the paper. PvdH, FWA and DLO provided feedback on written work.

contains fewer categories. In this study, we benchmark the state-of-the-art ICD classification systems and two baseline systems on a large data set constructed from Dutch cardiology discharge letters at UMCU hospital. Performance of all systems is evaluated for both the easier chapter-level ICD codes and single-label version of the task found in the literature, as well as for the lower-level ICD hierarchy and multi-label task that is needed in practice. We find that state-of-the-art methods outperform the baseline for the single-label version of the task only. For the multi-label task, the baselines are not defeated by any state-of-the-art system, with the exception of HA-GRU, which does perform best in the most difficult task on accuracy. We conclude that practical performance may have been somewhat overstated in the literature, although deep learning techniques are sufficiently good to complement, though not replace, human ICD coding in our application.

5.1 Introduction

ICD-10 is the 10th edition of the International statistical Classification of Diseases, a repository maintained by the World Health Organization to provide a standardized system of diagnostic codes for classifying diseases (Atutxa, de Ilarraza, Gojenola, Oronoz, & Perez de Viñaspre, 2019; Baumel, Nassour-Kassis, Cohen, Elhadad, & Elhadad, 2018). These classification codes are vastly used in clinical research and are a part of the electronic health records (EHRs) in the University Medical Center Utrecht (UMCU), the Netherlands. Currently, the task of assigning classification categories to the diagnoses is carried out manually by medical staff. Manual classification of diagnoses is a labor-intensive process that consumes significant resources. For this reason, a number of systems have been proposed to automate the disease coding process with machine learning algorithms trained on data generated by medical experts.

The ICD coding task is challenging due to the use of free-text, multi-label setting of diagnosis codes and the large number of codes (Atutxa et al., 2019; Boytcheva, 2011). Several attempts have been made to automatically assign ICD codes to medical documents, ranging from rule-based (Baghdadi et al., 2019; Boytcheva, 2011; Koopman, Karimi, et al., 2015; Nguyen et al., 2018) to machine learning approaches (Atutxa et al., 2019; Baumel et al., 2018; L. Cao, Gu, Ni, & Xie, 2019; Y. Chen, Lu, & Li, 2017; Du et al., 2019; Duarte et al., 2018; Karimi, Dai, Hassanzadeh, & Nguyen, 2017; Kemp, Rajkomar, & Dai, 2019; Koopman, Zuccon, Nguyen, Bergheim, & Grayson, 2015; Lin et al., 2019; Liu et al., 2018; Miranda, Martins, Silva, Silva, & Leite, 2018; Mujtaba et al., 2017; Zweigenbaum & Lavergne, 2016; Mullenbach et al., 2018; Nigam, 2016; Pakhomov, Buntrock, & Chute, 2006; Shing, Wang, & Resnik, 2019; Xie, Xiong, Yu, & Zhu, 2019). Rule-

based methods have good performance when: (1) the terms to be categorized follow regular patterns, (2) the number of ICD labels is quite small, and (3) the task is limited to single-label classification (Atutxa et al., 2019). Unfortunately, with ICD classification these conditions seldom apply.

When a coded data set is available and the range of the ICDs to label is large, machine learning based techniques have been successful (Atutxa et al., 2019; Baumel et al., 2018; L. Cao et al., 2019; Duarte et al., 2018; Miranda et al., 2018; Nigam, 2016). An approach (Boytcheva, 2011) for automatic matching of ICD-10 classification of Bulgarian free text was based on support vector machines (SVM). Zweigenbaum and Lavergne (2016) suggested a hybrid method for ICD-10 coding of death certificates based on a dictionary projection method and a supervised learning algorithm. They used the systemic nomenclature of medicine (SNOMED) and the unified medical language source (UMLS) to set up the dictionary projection method. Koopman, Zuccon, et al. (2015) trained 86 SVM classifiers to identify cancers, first identifying the presence of a cancer by one classifier and later in a cascaded architecture classifying the cancer type according to ICD-10 codes using 85 different SVM classifiers.

Recently, deep learning methods boosted benchmarked results in various text mining studies (Gargiulo, Silvestri, & Ciampi, 2018; Shickel, Tighe, Bihorac, & Rashidi, 2017; Kalyan & Sangeetha, 2020; Xiao et al., 2018), including in automated ICD coding (Atutxa et al., 2019; Baumel et al., 2018; Du et al., 2019; Duarte et al., 2018; Karimi et al., 2017; Lin et al., 2019; Liu et al., 2018; Miranda et al., 2018; Mujtaba et al., 2017; Mullenbach et al., 2018; Nigam, 2016; Shing et al., 2019). Karimi et al. (2017) described a deep learning method for ICD coding, reporting on tests over a data set of radiology reports. The authors proposed to use a convolutional neural network (CNN) architecture, attempting to quantify the impact of using pre-trained word embeddings for model initialization. The best CNN model outperformed baseline SVM, random forest, and logistic regression models using bag-of-words (BOW) representations. BOW is a vector representation method, demonstrating each document by one vector of features, i.e. words or combinations of words (n-grams). In (Nigam, 2016), recurrent neural networks (RNNs) have been applied to the multi-label classification task for assigning ICD-9 labels to medical notes, finding that an RNN with long short-term memory (LSTM) units shows an improvement over the binary relevance logistic regression model. Atutxa et al. (2019) evaluated different architectures of neural networks for multi-class document classification as a language modeling problem. In their experiments, the results of ICD-10 coding using the RNN-CNN architecture outperformed alternative approaches. Baumel et al. (2018) investigated four models namely SVM, continuous-BOW (CBOW), CNN and hierarchical attention bidirectional gated recurrent unit (HA-GRU) for attributing multiple ICD-9 codes. The HA-GRU model achieved the best performance. A drawback of the existing literature is that the performance of different systems is difficult to com-

Table 5.1: UMCU data set.

Feature	Description
Taxonomy	ICD-10
Language	Dutch
Nb of records	5,548
Nb of unique tokens	148,726
Avg nb of tokens / records	936
Nb of full labels	1,195
Nb of rolled-up labels	608
Label cardinality	4.7
Label density	0.0039
% labels with 50+ records	8.03

pare, because the ICD classification task is often made easier by only considering the top-level "chapters" of the ICD hierarchy, or by only considering a single label as the output.

In the current application, we sought to implement a system to support human ICD coding of Dutch-language discharge letters at UMCU hospital. We explicitly aim at multi-label classification of three-digit ICD-10 codes, a task that is relatively difficult. Here, we present a benchmark of five state-of-the-art systems, all deep learning models, and two baseline methods based on BOW and pretrained embeddings with SVM. We aim to evaluate both the relative performance of these systems, which were all reported to outperform others, as well as the overall level of performance for potential support of human ICD coding, using a data set of UMCU cardiology discharge letters.

5.2 Methods

5.2.1 Case Study

Table 5.1 provides the characteristics of the data set of discharge letters collected at the department of Cardiology in the UMCU. A hospital discharge letter is a medical text summary describing information about patient's hospital admission and treatments. UMCU cardiology discharge letters are coded based on the ICD-10 of cardiovascular diseases.

ICD-10 has a hierarchical structure, connecting specific diagnostic codes through is-a relations . The hierarchy has several levels, from less specific to more specific. ICD codes contain both diagnosis and procedure codes. In this paper, we focus on diagnosis codes. ICD-10 codes consist of three to seven characters. For example, I50.0 shows the "congestive heart failure" disease, and I50 is its rolled-up code that shows the heart failure category in chapter IX: "Diseases

of the circulatory system".

In Table 5.1, cardinality is the average number of codes assigned to records in the data set. Density is the cardinality divided by the total number of codes. We filtered out ICD codes with less than 50 observations on their frequency. We note that there are approximately 64 frequent labels with at least 200 records in UMCU data set. ICD codes in this data set are mainly from chapters 4, 9, and 21. Figure 5.1 illustrates the ICD rolled-up codes with more than 400 appearance in the UMCU data set. I25, Z95, I10, I48 and I50 are the top frequent rolled-up codes (at least 1000 counts) in our data set.

In this study, we experimented with two versions of the label set: one with the 22 ICD chapters and one with the labels rolled up to their three-digit equivalent.

5.2.2 Preprocessing

Preprocessing the data set of discharge letters comprises the following steps: (i) we anonymize the letters for legal and privacy reasons. We use $DEDUCE$ (Menger et al., 2018), a pattern matching tool for automatic de-identification of Dutch medical texts; (ii) we use the tm (Feinerer, 2019) and *tidytext* (Silge & Robinson, 2016) packages in R to trim whitespace, remove numbers, and convert all characters to lower case; (iii) we tokenize all texts using the $Pythonscikit - learn$ (Pedregosa et al., 2011) feature extractor, *gensim* library (Rehurek & Sojka, 2010) and the tokenizer in the $Keras$ library (Chollet et al., 2015).

5.2.3 Classification Methods

To employ the classification methods, we investigate two methods of vector representation:

- Bag-of-words (BOW; baseline)
- Word embeddings (average word vectors)

We use SVMs with each of the vector representations. We also assess the following neural network architectures for the automatic ICD coding of the Dutch discharge letters.

- CNN
- LSTM and BiLSTM
- HA-GRU

With these deep learning architectures, the first layer is the word embedding layer to represent patients' discharge letters. Hyperparameters of the models are formulated on the corresponding cited studies, while we tuned some based on the development set using a random parameter search.

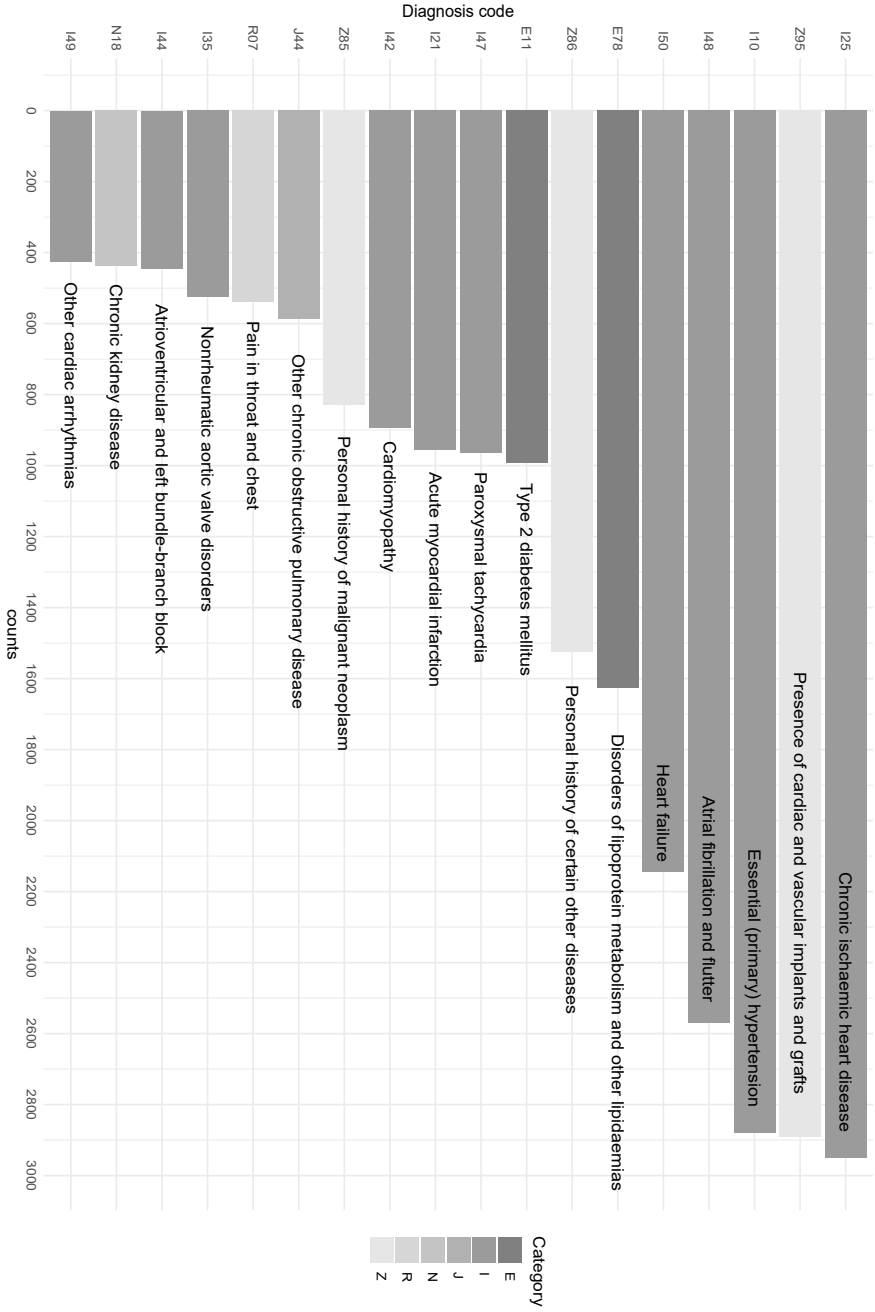


Figure 5.1: ICD rolled-up codes with more than 400 appearances in the UMCU data set.

5.2.3.1 Baseline: Support Vector Machines Using Bag-of-Words

We use a one-vs-all, multi-label binary SVM classifier as the baseline learning method for ICD-10 classification. Baghdadi et al. (2019), Koopman, Karimi, et al. (2015), Mujtaba et al. (2017), and Boytcheva (2011) applied SVM classifers for the task of ICD coding. We calculate the BOW representations using the preprocessed discharge letters. We also use the TFiDF vectorizer. The baseline model fits a one-vs-all binary SVM classifier with linear kernel for each ICD code against the rest of the codes.

5.2.3.2 Word Embeddings: Support Vector Machines Using Average Word Vectors

Word embeddings (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013) are vector representations for texts, representing words by capturing similarities between them (for a recent review on word embeddings in clinical natural language processing see (Kalyan & Sangeetha, 2020)). Skip-gram and CBOW are two ways of learning word embeddings. Both approaches use a simple neural network to create a dense representation of words. The CBOW tries to predict a word (target word) from the words that appear around it (context), while skip-gram inverts contexts and targets, and tries to predict context from a given word. Baumel et al. (2018) examined the word embedding representations for ICD coding and achieved better scores comparing to the BOW representations. In this study, we train CBOW word embeddings in *gensim*. We set the vector dimensionality to 300, the window size to 5, and discard the words that appear only once in the training set. We then use the average of word embeddings to represent each discharge letter. These embeddings are then inputs to the classification model defined by the baseline SVM.

5.2.3.3 Convolutional Neural Networks

To be able to capture the order of the words as well as multi-word expressions, the next model we investigate is a CNN model. CNN has proven to be a good method for text classification and is also applied for the task of ICD coding (Baumel et al., 2018; Du et al., 2019; Karimi et al., 2017). The CNN represents texts at different levels of abstraction, essentially choosing the most salient n-grams. We perform one dimensional convolutions on the embedded representations of the words. The architecture of this model is very similar to the average word embeddings model, but instead of averaging the embedded words we apply a one dimensional convolution layer with filter f, followed by a max pooling layer. One dimensional convolution layers have proven effective for deriving features from sequences data (Du et al., 2019). In our experiments, we used the same embedding parameters as in the average word embeddings model. In addition, we set the

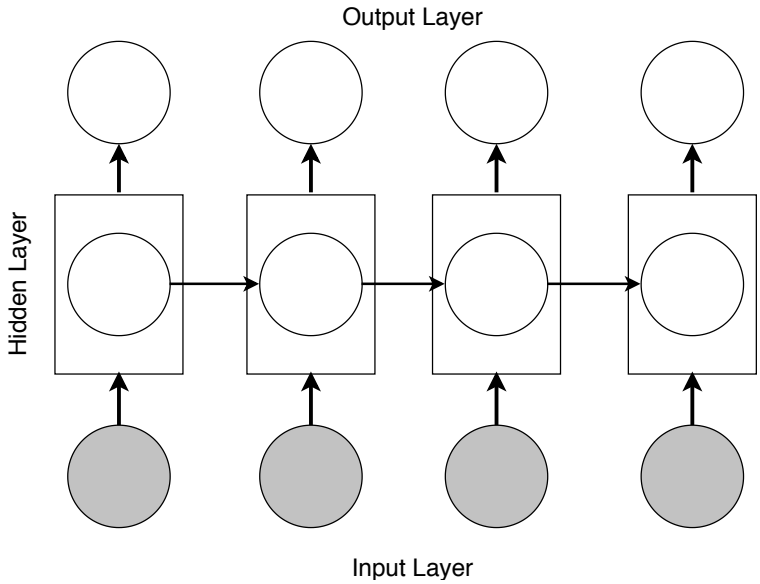


Figure 5.2: RNN architecture overview.

number of filters to 128, and the filter size to 5. On the output of the max pooling layer, a fully connected neural network (two dense layers) was applied for the classification of the ICD-10 codes. The hidden dense layer contains 128 units and uses the relu activation function, and the output layer uses a softmax function to determine if the ICD code should be assigned to the letter. We also examine the CNN model with two convolution layers and two max pooling layers. In this setting, we employed a dropout layer after the first max pooling layer with rate 0.15.

5.2.3.4 Long Short-Term Memory and Bidirectional Long Short-Term Memory

Feed-forward neural networks require fixed length contexts that need to be specified ad hoc before training (Chung, Gulcehre, Cho, & Bengio, 2014). For automated ICD coding, this means that neural networks see relatively few preceding words when predicting the next one. RNNs avoid this problem by not consuming all the input data at once (Chung et al., 2014; Mikolov, Karafiát, Burget, Cernocky, & Khudanpur, 2010; Miranda et al., 2018). An RNN is a straightforward adaptation of the standard feed forward neural network to allow it to model sequential data (Hochreiter & Schmidhuber, 1997; Sutskever, Martens, & Hinton, 2011). At each time-step, the RNN receives an input, updates its hidden state, and makes a prediction (see Figure 5.2).

By using recurrent connections, information can cycle inside these networks for

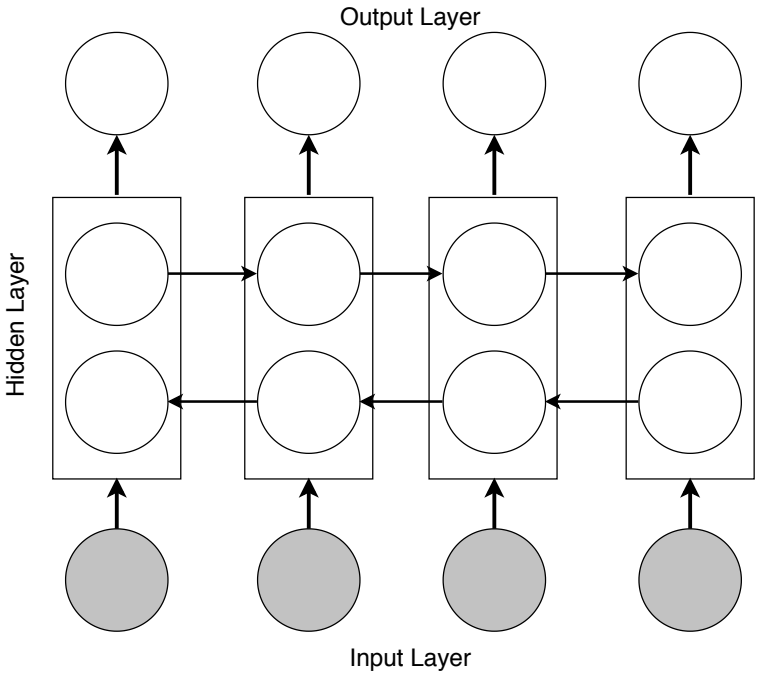


Figure 5.3: BiLSTM architecture overview.

an arbitrarily long time. LSTM models are variants of RNNs with memory gates that take a single input word at each time step and update the models' internal representation accordingly (Hochreiter & Schmidhuber, 1997). RNN is extended to use LSTM units, simply replacing the nodes in hidden layers in Figure 5.2 with LSTM units.

To overcome the limitations in RNNs using all available input information in the past and future of a specific time frame, bidirectional LSTM (BiLSTM) model is introduced by Schuster and Paliwal (1997). The BiLSTM model as shown in Figure 5.3 is an extension of the RNN model using LSTM units, that combines two LSTMs with one running forward in time and the other running backward. Thus the context window around each word consists of both information prior to and after the current word.

RNN models have been applied extensively on textual data for natural language processing, as well as in the medical domain and ICD coding (Atutxa et al., 2019; Baumel et al., 2018; Du et al., 2019; Duarte et al., 2018; Miranda et al., 2018; Nigam, 2016).

In this study, we used the *Keras* library to implement RNN models for automated ICD coding. We implemented LSTM and BiLSTM. We keep the same embedding parameters as in the average word embeddings model. We experimented with RNN models directly on the word sequence of all the discharge letters. How-

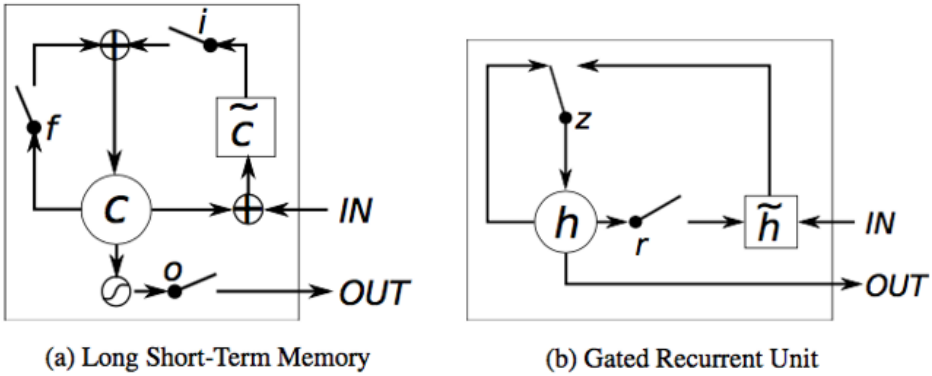


Figure 5.4: (a) LSTM memory cell: c is the memory cell, $\tilde{c}$ is the new memory cell content. i, f and o are the input, forget and output gates, respectively. (b) h and $\tilde{h}$ are the activation and candidate activation, respectively. r and z are the reset and update gates.

ever, as in previous studies on textual data, the fact that our data contains long texts creates a challenge for preserving the gradient across thousands of words. Therefore, we used dropout layers to mask the network units randomly during the training (Gal & Ghahramani, 2016). We set the number of hidden units in the RNN layers at 100. Dropout and recurrent dropout were added to avoid overfitting, both at a 0.2 rate. On the output of the recurrent layer, a fully connected neural network with the setting in CNN was applied for classification of the ICD-10 codes.

5.2.3.5 Hierarchical Attention Bidirectional Gated Recurrent Unit

GRU can be considered as a variation on the LSTM, that is a gating mechanism in RNN (Figure 5.4) aims to solve the vanishing gradient problem (Cho, Van Merriënboer, Gulcehre, et al., 2014). Figure 5.4 compares the memory cell structures of the LSTM and the GRU.

The GRU has a slightly different architecture where it combines both the input (gate i) and forget (gate f) gates into a single gate called the update gate (gate z). Also, it merges the cell state and the hidden state. This results to a reduced number of parameters as compared to LSTM architecture and in some cases has resulted in faster convergence and a more generalized model (Duarte et al., 2018).

Baumel et al. (2018) proposed a HA-GRU model with label-dependent attention layer to classify diseases codes. Since the GRU model is too slow when applied to long documents as it requires as many layers as of the document length, they developed a HA-GRU to be able to handle multi-label classification. In this paper, we implemented the HA-GRU (Baumel et al., 2018) for the ICD-10 classification of cardiovascular diseases. The HA-GRU is a hierarchical model with two levels of

bidirectional GRU encoding. The first bidirectional GRU operates over tokens and encodes sentences. The second bidirectional GRU encodes the entire document, applied over the encoded sentences. In this architecture, each GRU is applied to a much shorter sequence compared with a single GRU.

We applied the HA-GRU model using the *Dynet* deep learning library (Neubig et al., 2017) for ICD coding. The attention mechanism in the HA-GRU has the advantage that each label is invoked from different parts of the text. This allows the model to focus on the relevant sentences for each label (Choi, Bahadori, Schuetz, Stewart, & Sun, 2016). As for our previous deep learning models, we kept the same embedding parameters in the average word embeddings model. We used a neural attention mechanism with 128 hidden units to encode the bidirectional GRU outputs. The first GRU layer encoded the sentences into a fixed length vector. Then the second bidirectional GRU layer uses 128 attention layers to generate an encoding specific to each class. Finally, we applied a fully connected layer with *softmax* activation.

5.2.4 Evaluation Measures

Two evaluation measures are considered: accuracy, and F1 score. In the single-label classification scenario, accuracy is the fraction of correctly classified discharge letters to the whole collection of discharge letters. F1 is the harmonic mean of the fraction of positively coded discharge letters and the fraction of actual discharge letters that are positively classified. Accuracy is a simple and intuitive measure, yet F1 takes both false positives and false negatives into account. F1 score is a good measure for the ICD classification task as this task has a large number of categories and usually contains imbalanced data. To evaluate the multi-label classification performance, we use the following sample-based metrics for accuracy and F1:

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{5.1}$$

$$\text{F1} = \frac{1}{n}\sum_{i=1}^{n}\frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \tag{5.2}$$

Where:

Y_i = set of predicted ICD codes
Z_i = set of ground truth ICD codes
n = number of samples

We evaluate our experimental results in two scenarios:

- single-label prediction: a model assigns one label to each patient letter;

Table 5.2: Single-label performance: accuracy and F1 score on two settings (ICD chapters and rolled-up ICDs) for the models when trained on the UMCU discharge letters.

Method	ICD chapters		Rolled-up ICD codes	
	Accuracy	F1	Accuracy	F1
BOW SVM (baseline)	54.8	54.8	14.1	14.1
Average word embeddings (SVM)	54.9	**54.9**	18.2	18.2
CNN(1conv)	57.3	49.2	22.1	17.4
CNN(2conv)	59.2	54.0	22.5	18.1
LSTM	73.0	38.1	19.1	14.1
BiLSTM	**73.9**	41.3	23.2	**21.8**
HA-GRU	72.5	43.5	**23.7**	19.8

- multi-label prediction: a model assigns multiple labels per patient letter.

5.3 Results

We used the *train-test* split function from the model selection module implemented in the *scikit-learn* library to randomly split the data set into train and test sets. We separate 25% of the data as the test set and the rest as for training. To evaluate the proposed models on the data set of cardiovascular discharge letters, we conducted the following experiments. In the first setting, we trained the models on the training set separately using chapters as the labels. All models were evaluated on the test set according to the evaluation measures. In the second setting, we only considered the rolled-up ICD-10 codes to their three-digit codes.

5.3.1 Single-label Prediction Performance

Table 5.2 presents the obtained results for each model for both experimental settings (ICD chapters and rolled-up ICD codes) on the single-label scenario. In this case, a single code is predicted for every testing patient's letter. Bolded values in Table 5.2 indicate the best-performing model for each category.

BiLSTM gives the best accuracy in the ICD-10 chapters i.e. 73.9%, while the SVM classifier using the average word embedding has the highest F1 score of 54.9%. HA-GRU gives the best accuracy results in the rolled-up ICD-10 setting i.e. 23.7%, while the BiLSTM model has the highest value in F1 score with 21.8%.

Table 5.2 shows that the difference between the results of the rolled-up ICDs and the ones for the chapters is considerable. This is expected given the large number of the rolled-up ICD codes comparing to the number of the ICD chapters. We note that the SVM classifier is still competitive with the deep learning architectures in our application.

Table 5.3: Multi-label performance: accuracy and F1 score on two settings for the models when trained on the UMCU discharge letters.

Method	ICD chapters		Rolled-up ICD codes	
	Accuracy	F1	Accuracy	F1
BOW SVM (baseline)	**62.3**	**74.3**	11.6	20.2
Average word embeddings (SVM)	60.4	72.6	12.5	**25.8**
CNN(1conv)	38.1	46.3	09.0	16.1
CNN(2conv)	42.2	49.0	12.4	19.1
LSTM	53.4	59.6	11.7	18.8
BiLSTM	55.0	70.1	13.7	23.2
HA-GRU	56.8	71.3	**15.9**	24.3

5.3.2 Multi-label Prediction Performance

Table 5.3 presents the results for the multi-label task. In this scenario, corresponding to the prediction made by the classification models, every ICD label that presents a probability above a defined threshold is considered as a predicted output code. We assign the threshold in such a way that the label cardinality for the test set is in the same order as the label cardinality in the training set. Bolded values in Table 5.3 indicate the best-performing model for each category.

For the multi-label scenario, the SVM classifier gives the best results in F1 score for the chapter labels and for the rolled-up codes with values equal to 74.3% and 25.8%, respectively. The former is the F1 score for the BOW representation and the latter is the one for the word embeddings. In terms of accuracy, when the number of ICDs to be coded are large the HA-GRU has the best results with 15.9%.

By comparing Table 5.2 and Table 5.3, it is notable that the difference between the results on chapters and the results on the rolled-up codes is more consistent when we applied the CNN models using our case study. With regard to the single-label task, CNNs have the highest values of F1 of about 54% and 18.1%, respectively, for the ICD chapters and the rolled-up codes. For the multi-label task these values are equal to 49% and 19.1%.

5.4 Discussion

Automated ICD-10 classification can potentially save valuable time and resources in a clinical setting. In this study, we compared several state-of-the-art ICD coding systems on a data set of Dutch-language discharge letters.

Classification performance of the 22 higher-level codes is very promising, especially when only a single label is considered. For this version of the task, RNNs (LSTM, BiLSTM, and HA-GRU) showed good performance, as reported in the

literature. However, in many practical applications, including our own, a lower level of classification is required, and each letter receives multiple ICD codes. For this version of the task, performance was somewhat disappointing, and state-of-the-art systems failed to outperform the baseline BOW SVM with linear kernel. An exception is the HA-GRU system, which had the best accuracy, and showed an F1 performance close to that of the baseline.

While none of the systems were able to achieve a level of classification accuracy on the most difficult versions of the ICD classification task that would allow them to completely *replace* a human coder, they do show performance that is good enough to *suggest* codes in an interaction with the human. Future work could investigate the performance of human-in-the-loop systems, for example by employing active learning.

A question that may arise is whether machine learning could be supplanted with a rule-based system. This is possible for the higher-level codes using information retrieval and natural language processing methods (Pakhomov et al., 2006). However, developing rule-based systems with manually coded rules is tremendously difficult for the lower levels of ICDs. There are a large number of ICD codes in lower levels of the ICD hierarchy, and a small number of observations per ICD code. Deep learning-based models are useful here because they obviate the need for manual feature engineering (Atutxa et al., 2019). For this reason, we believe machine learning remains an attractive alternative to rule-based systems.

A second consideration is the question of model interpretability. Here, the deep learning models that form the current state of the art are especially challenging in this regard, and this may be a point in favor of "simpler" methods such as BOW: the more opaque the model, the less willing clinicians may be to accept artificial intelligence recommendations. Although it is not clear whether this is a problem for ICD-10 coding specifically, future work could focus on developing more interpretable systems or generic prediction explanation methods that mitigate this problem. Moreover, such systems could be very powerful when combined with a human-in-the-loop approach, by allowing the human to learn how text can be written to teach the correct code to the system.

Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest.

Chapter 6

Multi-label Detection of ICD-10 Codes in Cardiology Discharge Letters using Neural Networks

Sammani, A.[1], Bagheri, A.[1], Van der Heijden, P. G. M., Te Riele, A. S. J. M., Baas, A. F., Oosters, C. A. J., Oberski, D. L., & Asselbergs, F. W. (2020, submitted). Multi-label Detection of ICD-10 Codes in Cardiology Discharge Letters using Neural Networks.

Abstract

Standard reference terminology of diagnoses and risk factors is crucial for billing, epidemiological studies and inter/intranational comparisons of diseases. The International Classification of Disease (ICD) is a standardized and widely used method, but manual classification is an enormously time-consuming endeavour. Natural language processing together with machine learning allows automated structuring of diagnoses using ICD-10 codes. Limited performance of machine learning models, the necessity of gigantic data sets and poor reliability of terminal parts of these codes restrict clinical usability. We aimed to create a high performing pipeline for automated classification of reliable ICD-10 codes in free medical text in cardiology. We focused on frequently used and well defined three-digit

[1]Shared first authorship

Author contributions: AS and AB designed the study. AB developed the statistical methods and source code. AS and AB analyzed the data and experiments. AS and AB wrote the paper. PvdH, AtR, AFB, CAJO, DLO and FWA provided feedback on written work.

ICD-10 codes that still have enough granularity to be clinically relevant such as atrial fibrillation (I48) or acute myocardial infarction (I21). Our pipeline uses a deep neural network called Bidirectional Gated Recurrent Unit Neural Network and was trained with 5,548 discharge letters. As in clinical practice discharge letters may be labeled with more than one ICD-10, we assessed the single- and multi-label performance of main diagnoses and cardiovascular risk factors. We investigated using both the entire body of text and only the summary paragraph, supplemented by age and sex. Given the privacy sensitive information included in discharge letters we added a de-identification step. Multi-label performance was high, with a sensitivity of 88%, specificity of 98%, and positive and negative predictive values of 93% and 95%. Using solely the summary paragraph of discharge letters decreased single-label negative predictive values for cardiovascular risk factors from 75% to 46%. Adding variables age / sex did not affect results. Because of its high performance, this pipeline can be useful to the decrease administrative burden of classifying discharge diagnoses and may serve a scaffold for reimbursement and research applications.

6.1 Introduction

Electronic health records enable fast information retrieval and contain both structured (e.g. laboratory values, numeric measurements) and unstructured data (free text in clinical notes) (Jensen et al., 2012). Clinical discharge letters are an important source of information, but the translation from free text to structured data remains challenging (Bagheri et al., 2020a). To structure diagnoses, the international classification of diseases (ICD-10) coding system was created. This classification system is hierarchical and multiple codes may be assigned to a single discharge letter (multi-label). ICD-10 is alphanumerically structured, with seven possible digits arranged hierarchically as shown in Figure 6.1 (Hirsch et al., 2016). The classification is performed by practitioners, managers or medical coders and serves worldwide in clinical practice (e.g. medical history and billing), research (e.g. trial recruitment) and (inter)national epidemiological studies (Jensen et al., 2012; Bagheri et al., 2020a; Hirsch et al., 2016; Atutxa et al., 2019; Stausberg, Lehmann, Kaczmarek, & Stein, 2008). Manual classification is an enormously costly endeavour, its quality depends on the profession of the coder and the reliability for terminal parts of ICD-10 codes is poor, even among trained medical coders (Stausberg et al., 2008).

Natural language processing (NLP) together with machine learning allows to automate ICD-10 coding for discharge letters (Bagheri et al., 2020a). This task is particularly challenging because of: (i) the unstructured nature of free text, (ii)

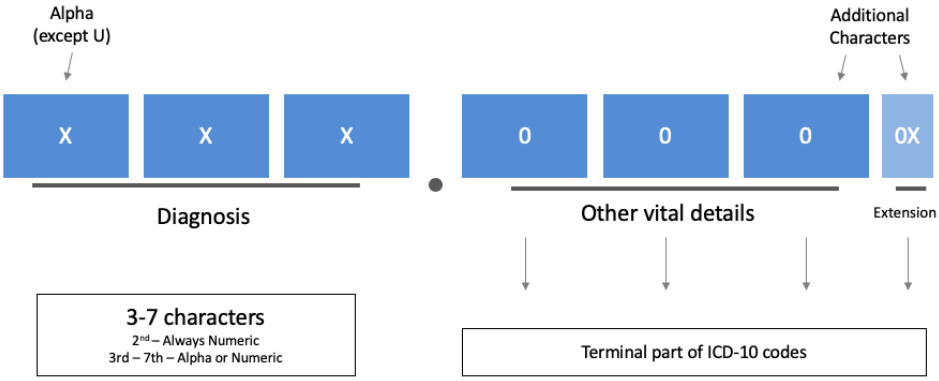


Figure 6.1: ICD-10 structure.

the multi-label setting of ICD10 codes and (iii) the large number of terminal ICD-10 codes (Atutxa et al., 2019). Several attempts have been made to automatically assign ICD-10 codes to medical documents ranging from rule-based to machine learning approaches (Bagheri et al., 2020a; Blanco et al., 2020). Generally speaking, rule-based methods have good performance, which is however restricted to conditions that seldomly occur in free-text clinical notes (given possibly ambiguous wording/spelling, multi-label classification and sparsity). Machine learning techniques on the other hand have shown increasingly promising results (Bagheri et al., 2020a; Atutxa et al., 2019; Blanco et al., 2020; Koopman, Karimi, et al., 2015). Supervised classification is often simplified by considering only top-level "chapters" of ICD-10 hierarchy or by only considering a single label as output. By doing so, these models do not depict a real-world situation and are less applicable in daily clinical practice (Atutxa et al., 2019; Koopman, Karimi, et al., 2015; L. Cao et al., 2019; Y. Chen et al., 2017; Du et al., 2019; Duarte et al., 2018; Karimi et al., 2017; Lin et al., 2019; Pakhomov et al., 2006; Perotte et al., 2014). Most recently, multi-label classification of detailed ICD-10 codes has been improved greatly with deep learning, showing better performance when using recurrent neural networks. These improved models however rely on enormous data sets (Bagheri et al., 2020a; Atutxa et al., 2019; Blanco et al., 2020). Poor reliability of terminal parts of ICD-10 codes combined with the limited performance of most models restricts clinical usability (Table 6.1) (Bagheri et al., 2020a; Atutxa et al., 2019; Koopman, Karimi, et al., 2015; L. Cao et al., 2019; Y. Chen et al., 2017; Du et al., 2019; Duarte et al., 2018; Karimi et al., 2017; Lin et al., 2019; Pakhomov et al., 2006; Perotte et al., 2014).

In this paper, we propose a pipeline for the automatic classification of ICD-10 codes from free-text clinical discharge letters from cardiology. We investigated the usage of solely the summary paragraph of discharge letters (conclusion), adding

Table 6.1: Performance of machine learning classifiers in literature.

Reference	F1 score	Classifier	Data set
(Atutxa et al., 2019)	0.84 – 0.95	RNN[a]	Death certificates from CépiDc (France), ISTAT (Italy) and a Hungarian database[b]
(Blanco et al., 2020)	0.70	RNN	Osakidetza Spanish basque public health system
(L. Cao et al., 2019)	0.68	HCAML[c]	Internal Chinese EHR[d] data set
(Y. Chen et al., 2017)	0.63	Longest Common Subsequence	ICD[e]-10 National Chinese data set
(Lin et al., 2019)	0.72	CNN[f]	Tri-service General Hospital Taipei data set with ICD-10 labels
(Du et al., 2019)	0.43	CNN	ICD-9 intensive care data set[g]
(Duarte et al., 2018)	0.65	Hybrid neural network	Cause of death autopsy reports (three-digit)
(Karimi et al., 2017)	0.81	CNN	ICD-9 radiology reports
(Koopman, Karimi, et al., 2015)	0.94	Binary SVM[h] classifier for four different codes	Australian Bureau of Statistics data set with ICD-10 cause of deaths%
(Pakhomov et al., 2006)	0.54	Naïve Bayes classifier	Random sample of HICDA[i] (A mayo-clinics adaptation of ICD-8) data set
(Perotte et al., 2014)	0.40	Hierarchy-based SVM	ICD-9 intensive care data set[j]

[a]RNN: Recurrent Neural Network [b]Using 128,000 training data [c]HCAML: Hierarchical Convolutional Attention for Multi-Label classification [d]EHR: Electronic Health Record [e]ICD: International Classification of Disease [f]CNN: Convolutional Neural Network [g]Using the same data set [h]SVM: Support Vector Machine [i]HICDA: Hospital Adaptation of the International Classification of Diseases [j]Using 447,336 training data and four codes ICD-10 codes to predict as outcome

Table 6.2: UMCU Cardiology data set.

Variable	Description
Taxonomy	International Classification of Disease - version 10
Language	Dutch
Number of unique records	5,548
Number of unique tokens	148,726
Average number of tokens per record	936
Number of rolled-up labels (i.e. I42)	608
Average number of codes per letter	4,7
% of labels with >50 letters	8,03%
Age. Median (IQs)	68 (1st: 58, 3rd: 77) years
Sex (% Female)	36% Female

clinical variables (age / sex) and multi-label classification as is the case in clinical practice. We focussed well-defined and frequently used three-digit ICD-10 codes related to cardiology with sufficient granularity to be clinically relevant such as atrial fibrillation (I48) or acute myocardial infarction (I21).

6.2 Results

6.2.1 Data Set

In total, 5,548 discharge letters from in-house cardiology patients were included in the data set with an average of 4.7 codes per letter (cardinality). Median age at discharge was 68 years (1st and 3rd quartiles [58-77]) and 36% of patients were female. Table 6.2 summarizes the characteristics and an example is given in Table 6.3. 64 different ICD-10 codes have at least 200 records in this data set. Most common ICD-10 code was I25 (chronic ischemic heart disease) followed by Z95, I10 and I48 (presence of cardiac vascular implants and grafts, primary hypertension and atrial fibrillation/flutter, respectively) with all at least 1000 individual counts.

6.2.2 Performance of Models

The performance of our best performing model, the Bidirectional Gated Recurrent Unit (BGRU), is summarized in Table 6.4, Table 6.5, and Table 6.6. Overall, performance was remarkably high for all selected ICD-10 codes. Performance was optimal using the entire corpus of the discharge letters, especially in terms of negative predictive value. Adding variables age and sex did not affect performance.

Table 6.3: An example letter from the UMCU Cardiology data set.

Example letter
Bovengenoemde patiënt was opgenomen op <*DATUM-1*> op de <*PERSOON-1*> voor het specialisme Cardiologie. **Reden van opname** STEMI inferior **Cardiale voorgeschiedenis**. Blanco **Cardiovasculaire risicofactoren:** Roken(-) Diabetes(-) Hypertensie(?) Hypercholesterolemie (?) **Anamnese**. Om 18.30 pijn op de borst met uitstraling naar de linkerarm, zweten, misselijk. Ambulance gebeld en bij aansluiten monitor beeld van acuut onderwandinfarct. **AMBU overdracht:**.500mg aspegic iv, ticagrelor 180mg oraal, heparine, zofran eenmalig, 3x NTG spray. HD stabiel gebleven...Medicatie bij presentatie.Geen.. **Lichamelijk onderzoek**. Grauw, vegetatief, Halsvenen niet gestuwd. Cor s1 s2 geen souffles.Pulm schoon. Extr warm en slank. **Aanvullend onderzoek**. AMBU ECG: Sinusritme, STEMI inferior III)II C/vermoedelijk RCA. Coronair angiografie. (...) .Conclusie angio: 1-vatslijden..PCI **Conclusie en beleid** Bovengenoemde <*LEEFTIJD-1*> jarige man, blanco cardiale voorgeschiedenis, werd gepresenteerd vanwege een STEMI inferior waarvoor een spoed PCI werd verricht van de mid-RCA. Er bestaan geen relevante nevenletsels. Hij kon na de procedure worden overgeplaatst naar de CCU van het <*INSTELLING-2*>. ..Dank voor de snelle overname. ..Medicatie bij overplaatsing. Acetylsalicylzuur dispertablet 80mg ; oraal; 1 x per dag 80 milligram ; <*DATUM-1*>. Ticagrelor tablet 90mg ; oraal; 2 x per dag 90 milligram ; <*DATUM-1*>. Metoprolol tablet 50mg ; oraal; 2 x per dag 25 milligram ; <*DATUM-1*>. Atorvastatine tablet 40mg (als ca-zout-3-water) ; oraal; 1 x per dag 40 milligram ; <*DATUM-1*> **Samenvatting** Hoofddiagnose: STEMI inferior wv PCI RCA. Geen nevenletsels. Nevendiagnoses: geen. Complicaties: geen Ontslag naar: CCU <*INSTELLING-2*>.

Table 6.4: Cardiovascular disease classification using only specific structured parts of discharge letters (conclusion/summary).

	E11	E78	I10	I21	I25	I42	I48	I50	N18	Z95
Sensitivity	0.88	0.77	0.70	0.97	0.81	0.92	0.84	0.88	0.94	0.76
Specificity	0.81	0.60	0.68	0.87	0.73	0.85	0.90	0.85	0.60	0.64
Pos Pred Value	0.99	0.97	0.78	0.98	0.88	0.99	0.95	0.95	0.99	0.86
Neg Pred Value	0.17	0.14	0.89	0.81	0.63	0.46	0.73	0.69	0.14	0.48
F1	0.93	0.86	0.74	0.97	0.84	0.95	0.89	0.91	0.97	0.81

Table 6.5: Cardiovascular disease classification using complete discharge letters.

	E11	E78	I10	I21	I25	I42	I48	I50	N18	Z95
Sensitivity	0.95	0.78	0.80	0.97	0.86	0.96	0.88	0.91	0.96	0.83
Specificity	0.79	0.67	0.79	0.86	0.78	0.91	0.90	0.82	0.79	0.76
Pos Pred Value	0.97	0.98	0.84	0.98	0.89	0.99	0.94	0.93	0.99	0.90
Neg Pred Value	0.66	0.15	0.74	0.82	0.74	0.73	0.80	0.78	0.44	0.64
F1	0.96	0.87	0.82	0.97	0.88	0.97	0.91	0.92	0.97	0.86

Table 6.6: Cardiovascular disease classification using complete discharge letters and variables age and sex.

	E11	E78	I10	I21	I25	I42	I48	I50	N18	Z95
Sensitivity	0.95	0.78	0.80	0.97	0.86	0.96	0.87	0.91	0.96	0.84
Specificity	0.80	0.69	0.77	0.86	0.77	0.90	0.89	0.81	0.78	0.76
Pos Pred Value	0.97	0.98	0.82	0.98	0.88	0.99	0.94	0.93	0.99	0.90
Neg Pred Value	0.68	0.15	0.74	0.83	0.74	0.74	0.79	0.78	0.42	0.65
F1	0.96	0.87	0.81	0.97	0.87	0.97	0.90	0.92	0.97	0.87

6.2.3 Main Diagnoses

Performance for main diagnosis (I21, I25, I42, I48, I50) was high (Tables 6.4, 6.5, 6.6). Main diagnoses are often present in the final summary paragraph of discharge letters. It is logical that our model performed relatively well using only this part of text. Sensitivity ranged from 84% for cardiomyopathy (I42) to 97% for acute myocardial infarction (I21). Specificity was 73% for chronic ischemic heart disease (I25) and highest for cardiomyopathy (I42) with 90%. NPVs ranged from 46% to 72% using only the summary paragraph. When using the entire corpus of discharge letters and adding the variables age and sex, the NPVs improved, ranging from 74% to 83%.

6.2.4 Risk Factors for Cardiovascular Disease and Renal Failure

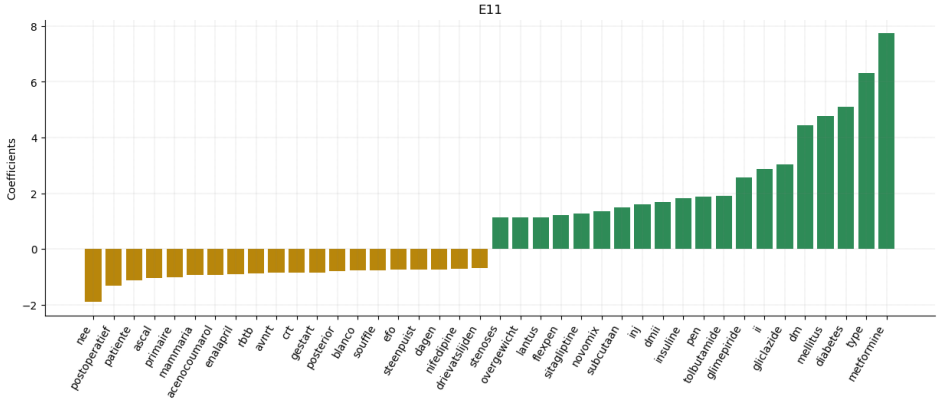
Performance of the model for cardiovascular risk factors (I10, E11, E78) and renal failure (N18) was high. PPVs range from 82% for hypertension (I10) to 98% for lipidemias (E78). Using only the summary paragraph however diminished the performance, especially in terms of NPV (68% to 17% for type 2 diabetes (E11) and 42% to 14% for renal failure (N18)). Using the entire corpus of the discharge letters resulted in higher negative predictive values and an increased performance overall.

6.2.5 Multi-label Classification of ICD-10 Codes

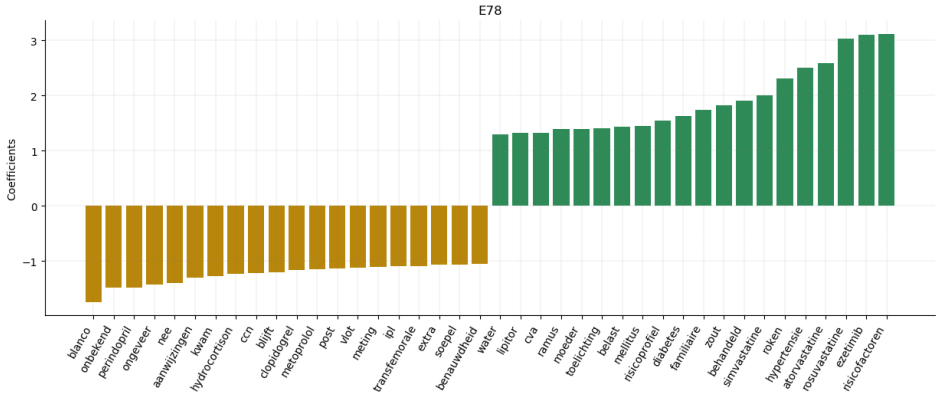
The performance of multi-label classification was high. Sensitivity was 88%, Specificity was 98%, PPV was 93%, NPV was 95% and the F1 score was 89%.

6.2.6 Word Coefficients

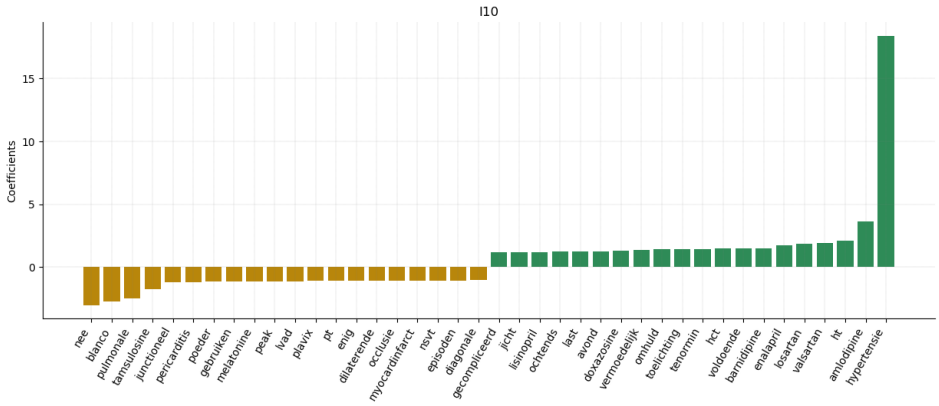
To interpret the model word coefficients have been plotted per ICD-10 code. Words that increase the prediction probability are delineated in green. For Type 2 diabetes (E11) these words are either related to the use of medication ("metformin", "gliclazide", "insulin"), are synonyms for E11 ("diabetes", "mellitus", "dmii") or are words that co-occur with cardiovascular risk factors ("overgewicht" (translation: overweight), "stenoses"). For hypertension (I10), the highest coefficients were reached with the synonyms and medication for hypertension as well ("hypertensie", "amlodipine", "valsartan", "ht"). This pattern can be seen for all ICD-10 codes. The words "blanco", "normale" and "nee" all have negative coefficients which illustrates the negative effect of these words in the ICD-10 codes E11, E78, I10, I21, Z95. The coefficients of all ICD-10 codes are visible in Figure 6.2.



(a)

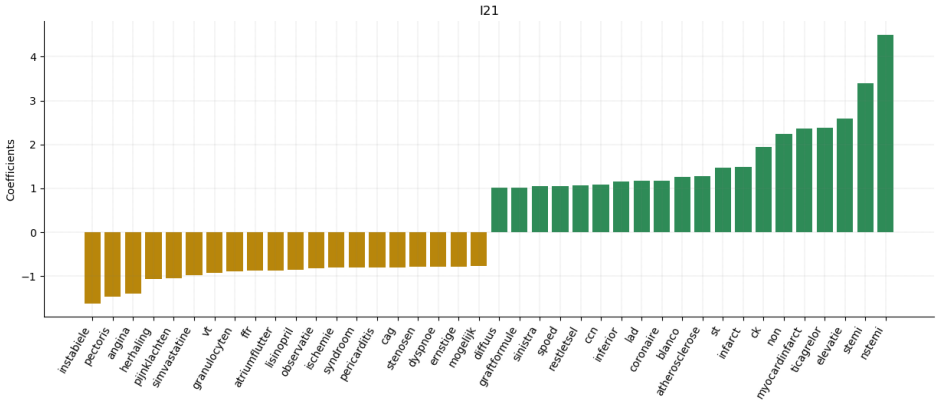


(b)

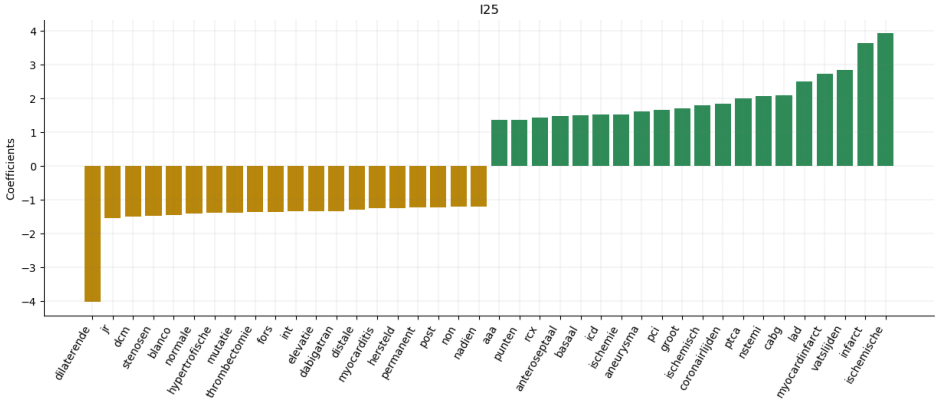


(c)

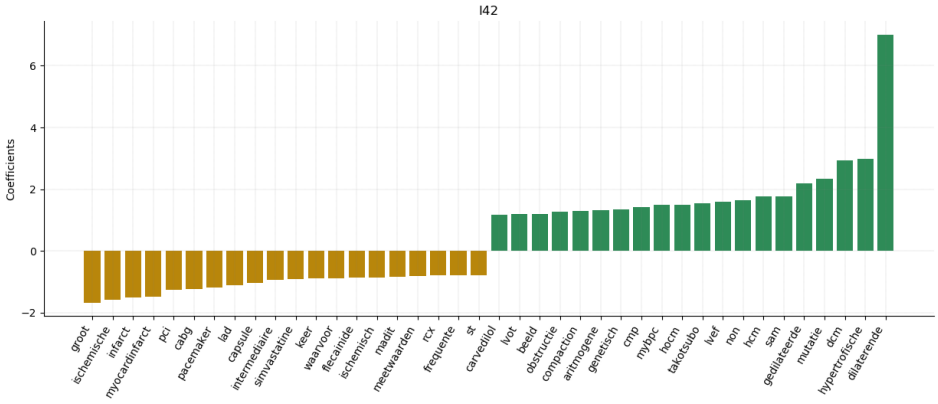
Figure 6.2: Word coefficients: (a) E11, (b) E78, (c) I10.



(d)

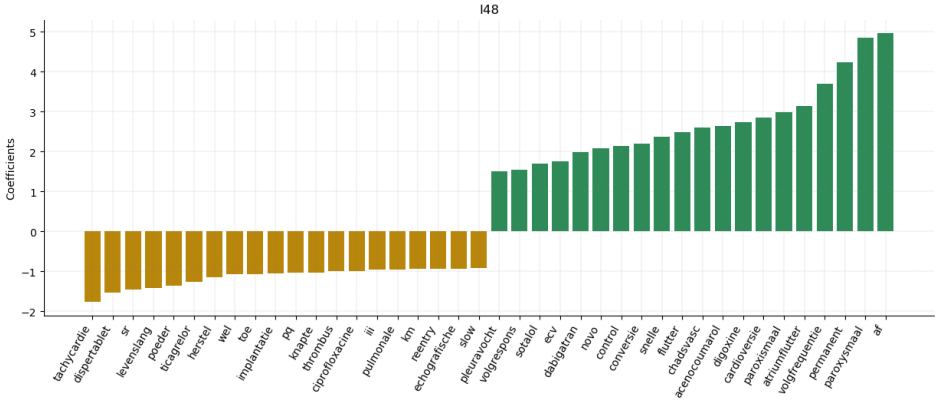


(e)

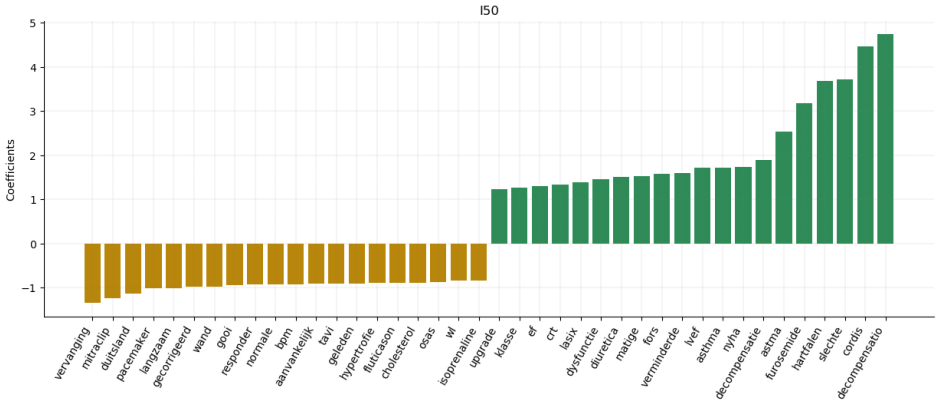


(f)

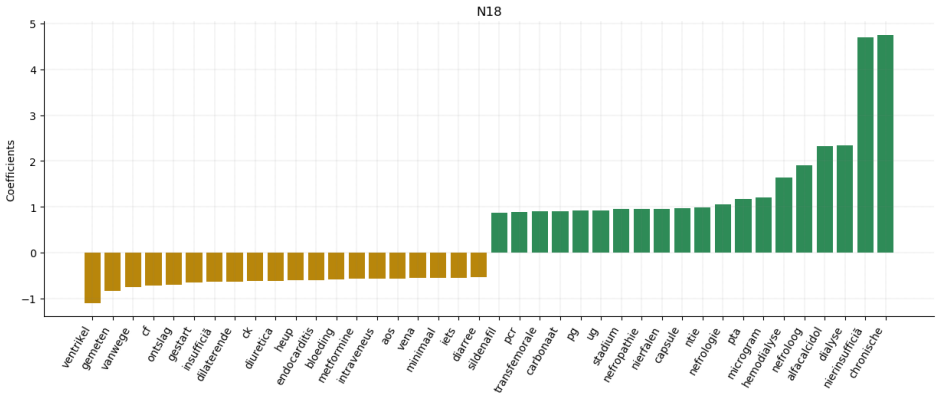
Figure 6.2: Word coefficients: (d) I21, (e) I25, (f) I42.



(g)

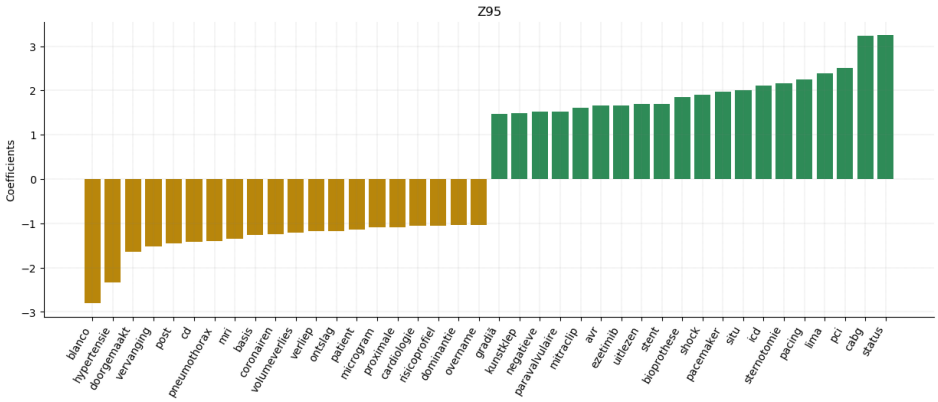


(h)



(i)

Figure 6.2: Word coefficients: (g) I48, (h) I50, (i) N18.



(j)

Figure 6.2: Word coefficients: (j) Z95.

6.3 Discussion

We created a deep learning pipeline for automatic multi-label ICD-10 classification in free medical text using Dutch cardiology discharge letters. Given the sensitive nature of these data, we included a de-identification step (Menger et al., 2018).

6.3.1 Prior Work

Prior work on NLP in cardiology was focused on specific relevant indicators such as hypertension, algorithms to identify Framingham heart failure signs and symptoms or identification of cardiovascular risk factors and outcomes (Sheikhalishahi et al., 2019). The use of RNN for cardiovascular diagnoses, risk factors and complications, however, remained relatively uncharted. Partially, this is due to rather low performance of some models limiting clinical usefulness (Koopman, Karimi, et al., 2015; L. Cao et al., 2019; Y. Chen et al., 2017; Du et al., 2019; Duarte et al., 2018; Karimi et al., 2017; Lin et al., 2019; Pakhomov et al., 2006; Perotte et al., 2014). Recent methodological developments in neural networks lead to high performing models, but they rely on limiting the number of codes (four) to predict, or require huge data sets of up to 128,000 training data points (Table 6.1) (Atutxa et al., 2019; Koopman, Karimi, et al., 2015). Limited performance of machine learning models, the necessity of gigantic data sets, and poor reliability of terminal parts of ICD-10 codes withhold them from replacing or aiding a human coder.

6.3.2 Proposed Model

In this work, we used a deep neural network and focused on clinical usefulness with both single and multi-label prediction in a relatively small data set of 5.548 clinical discharge notes. We extracted frequently used, well defined and clinically relevant three-digit ICD-10 codes (Stausberg et al., 2008). These three-digit codes still have enough granularity to include relevant diagnoses such as atrial fibrillation (I48) or acute myocardial infarction (I21). Next, we assessed and improved an already potent type of RNN (BGRU) by using semi-structured parts of text and adding clinical variables (age and sex). We then sought to explain our model using word coefficients. Even though our data set focused on cardiology, the pipeline is generalizable and may be trained with data from any other speciality.

6.3.3 High Performance with Bidirectional Gated Recurrent Unit Neural Network

A comparison of several state-of-the-art RNN ICD coding systems reported that classification performance is higher for ICD chapters than rolled-up codes. The previously reported F1 scores of ICD-10 chapters are around 50-60% at best and limited to 20-30% for rolled-up, more terminal codes (Bagheri et al., 2020a). BGRU has been promising for classification of medical text and prior experiments advocate either reducing granularity or increasing training data to improve performance (Bagheri et al., 2020a; Atutxa et al., 2019; Blanco et al., 2020). Additionally, the use of co-occurrences (association rule mining) for the initialization weights also positively impacted performance (Duarte et al., 2018). Unfortunately, in most settings training data are limited. Therefore, we tried reducing granularity of our data set whilst remaining clinically relevant without reducing the label-set size. By doing so, our pipeline reached F1 scores for rolled-up codes of 97%. Using the entire corpus of text rather than semi-structured parts also improved classification performance, especially for risk factors such as diabetes and hypertension that are seldomly mentioned in the summary paragraph of discharge letters. By building on prior work and using BGRU which is computationally less expensive, our reported performance is substantially higher than previously seen in smaller data sets, making it a useful and scalable tool for administrative and research support (Bagheri et al., 2020a; Atutxa et al., 2019; Duarte et al., 2018).

6.3.4 Clinical Usability

Most ICD-10 codes are used rarely in clinical practice, while a small amount of diagnoses comprise the majority of patients seen in cardiology clinics (Hirsch et al., 2016; Stausberg et al., 2008). To aid administrative support, our focus was directed towards multi-label classification. This resulted in a PPV of 93%, NPV of 95% and F1 score of 89% (Tables 6.4, 6.5, 6.6). We argue that this performance

is high enough to aid a human coder. From a clinical perspective, the high single label performance allows for patient identification in EHRs by using only the clinical discharge letters as a first step towards building research cohorts of interest. Less frequent ICD-10 codes, for rare diagnoses for instance, still require data sets large enough for machine learning and deep learning algorithms to perform well in ICD-10 classification (Bagheri et al., 2020a). For these diagnoses, rule-based methods may be a more viable option, given that the terms in text follow regular patterns and the task is limited to single-label classification (Atutxa et al., 2019). To accurately capture rare diagnoses, other more structured parts of the EHR may be useful such as laboratory results. A well-performing example is a simple classification algorithm for identification of patients with Systemic Sclerosis in the EHR by using positive antinuclear antibody titre thresholds (Jamian et al., 2019).

Automated coding system that combine simple classifiers with machine learning models are not new, as they have been successfully implemented in 2006 at the Mayo Clinic and resulted in an 80% reduction of staff engaged in manual coding (Pakhomov et al., 2006). More recently, a similar system for veterinary electronic health records (VetTag) was built, which classified veterinary clinical notes with diagnosis codes. Authors argue that processing these clinical notes has a tremendous impact on (veterinary) clinical data sciences (Zhang, Nie, Zehnder, Page, & Zou, 2019). Nonetheless, these promising results have not led to widespread use of automatic coding systems for discharge letters (Sheikhalishahi et al., 2019). It is clear that human coders can benefit by reviewing suggested ICD-10 codes rather than reading all discharge letters and translating them to proper ICD-10 codes (Pakhomov et al., 2006). Saved time can then be used to dive deeply into the correct terminal and detailed coding. However, there are two long-term concerns: the first is the actual implementation of these algorithms into software. Implementation is more than solely installing an automation pipeline. It requires new software which is embedded in existing workflows and prolonged maintenance. The second is the improvement of technology to suggest ICD-10 codes with high accuracy that are more complex and less frequent, which would require larger data sets and feedback algorithms. Further efforts may need to investigate implementation as well, rather than solely focusing on methodological fine-tuning.

6.3.5 Interpretability of the Neural Network

An important consideration is model interpretability. State-of-the-art deep learning models are challenging to grasp with no specialised knowledge in neural networks, and practice has shown that the easier the model, the wider its acceptance. There has been a significant increase in the use of machine learning methods but a notable proportion of works still use relatively simple methods: shallow classifiers, or combined with rule-based methods for higher interpretability (Sheikhalishahi et al., 2019). Interpretable results however may provide experts with supporting

evidence when confronted with coding decisions (Atutxa et al., 2019). We therefore attempted to provide insight into the model by using word coefficients. These results illustrate that synonyms of ICD-10 diagnoses or medication specifically prescribed for these diseases have the highest positive probabilities. Negative words (negation), such as "normal" or "no" decrease the probability of ICD-10 diagnoses, more noticeably for cardiovascular risk factors.

6.3.6 Conclusion

We propose a novel automated ICD-10 classifier BGRU pipeline with a de-identification step. Interpretation of the BGRU pipeline is made possible by using word coefficients. Because of its high performance, this pipeline can be useful to the decrease administrative burden of classifying discharge diagnoses and may serve a scaffold for reimbursement and research applications.

6.4 Methods

6.4.1 Medical Ethical Regulations and GDPR

This study was exempt from medical ethical regulations by the Medical Ethical Committee of the UMCU (no. 18-446). A data management plan was created and reviewed by the privacy security board to meet institutional and national requirements in the Netherlands for GDPR compliance.

6.4.2 Data Set

ICD-10 classification was retrieved from the electronic health records in the University Medical Center Utrecht (UMCU). ICD10 codes and discharge letters were available from the start of the electronic health record on 23-11-2018 until data extraction on 04-09-2019. We matched the letters to the ICD-10 classification by using patient ID and dates of admission/discharge from within the UMCU Research Data Platform. We removed ICD-10 codes with less than 50 observations (Sammani et al., 2019). Since the reliability of terminal codes is poor, simplification of ICD-10 codes is important to receive a valid image of healthcare reality (Stausberg et al., 2008). Selection of specific ICD codes was based on availability and clinical usability (sufficient granularity) of higher level rolled-up codes (e.g. I42 (cardiomyopathy) rather than I42.3 (endomyocardial (eosinophilic) disease)). The 10 selected codes are depicted in Table 6.7. and account for six main diagnoses (acute myocardial infarction, chronic ischemic heart disease, cardiomyopathy, atrial fibrillation/flutter, heart failure and presence of cardiovascular implant grafts) and four cardiovascular risk factors (type 2 diabetes, hyper/dyslipidemia, primary hypertension, chronic kidney disease).

Table 6.7: Selected three-digit ICD-10 codes.

ICD10 code	Description of code
E11*	type 2 diabetes mellitus
E78*	disorders of lipoprotein metabolism and other lipidemias
I10*	primary hypertension
I21	acute myocardial infarction
I25	chronic ischemic heart disease
I42	cardiomyopathy
I48	atrial fibrillation and flutter
I50	heart failure
N18*	chronic kidney disease
Z95	presence of cardiac and vascular implants grafts

* Risk factor for cardiovascular disease

6.4.3 Machine Learning Pipeline for ICD-10 Classification

The pipeline is summarized in Figure 6.3. Before feeding data into the machines different machine learning or deep learning algorithms, we first applied the following steps:

- We de-identified the letters using *DEDUCE* (Menger et al., 2018).
- We preprocessed the text (trimmed white-spaces, numbers and converted all characters to lowercase) using used the tm and *tidytext* packages in *R* (Jones, Maillardet, & Robinson, 2014).

To transform text into data a machine can understand (text representation), the output of our preprocessed text was then vectorized using word embedding. This method allows to represent words in such a way that it captures meanings, semantic relationships and context that words are used in. It is a dense feature representation in a low dimensional vector and has been proven to be a robust solution for most NLP issues. Word embedding is also the first layer in a neural network (NN) based classifier. After k-fold cross-validation ($k = 5$) we implemented a bidirectional gated recurrent unit neural network (BGRU).

6.4.4 Bidirectional Gated Recurrent Unit (BGRU) Neural Network

The general architecture of a BGRU model is shown in Figure 6.4. In this model, the input layer is the text from discharge letters and the output layer is the ICD-10 label. The model uses deep recurrent neural networks (RNN) in its hidden layers,

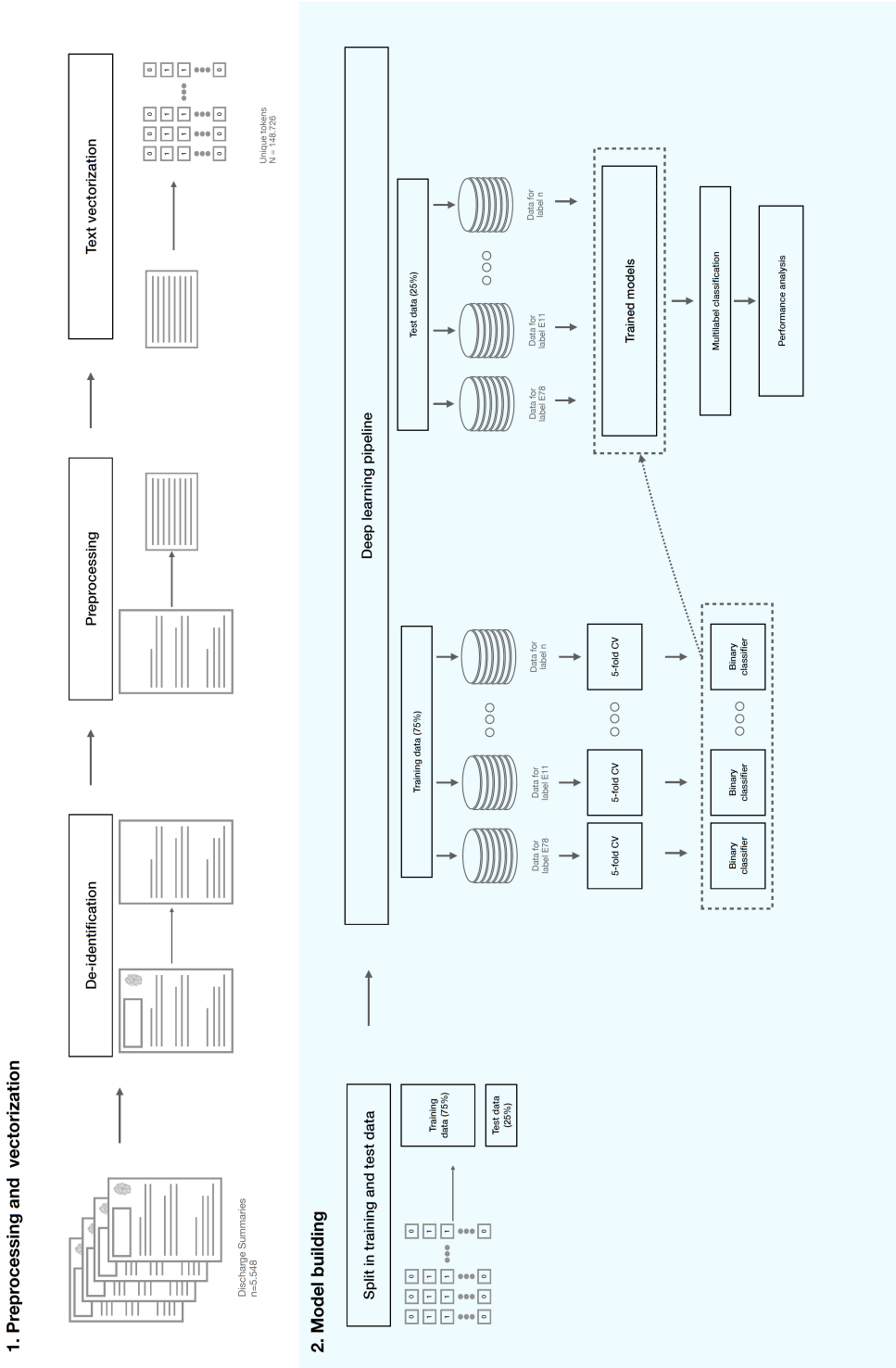


Figure 6.3: Summary of training and testing pipeline.

called gated recurrent units (GRUs). GRU is a type of RNN that can model sequential data. The GRU network receives an input at each timestep, updates its hidden state, and makes a prediction. By using recurrent connections, information can cycle inside these networks for an arbitrarily long time. However, RNNs are known to have difficulties learning the interactions between distant words because of long-range dependencies. This is known as the vanishing gradient problem. Extensions for neural networks, such as Long-Short Term Memory (LSTM) and GRU were specifically designed to combat this issue through a gating mechanism. Using GRUs also leads to a reduced number of parameters, faster convergence and a more generalizable model in comparison to other methods (Duarte et al., 2018).

We used the *Keras* library to implement the BGRU model for automated ICD-10 coding (Chollet et al., 2015). Vector dimensionality was set to 300, windows size to five and we discarded words that only appeared once in the training set. We experimented with the model directly on the word sequence of all the discharge letters. As in previous studies on textual data, the fact that our data contains long texts creates a challenge for preserving the gradient across thousands of words. Therefore, we used dropout layers to mask the network units randomly during the training (Gal & Ghahramani, 2016). We set the number of hidden units in the RNN layers at 100. Dropout and recurrent dropout were added to avoid overfitting, both at a 0.2 rate. On the output of the recurrent layer, a fully connected neural network (two dense layers) was applied for the classification of the ICD-10 codes. The hidden dense layer contains 128 units and uses the relu activation function, and the output layer uses a softmax function to determine if the ICD code should be assigned to the letter.

6.4.5 Assessment of Performance and Experiments

We investigated performance by randomly splitting the data set in a training (0.75) and testing (0.25) set. We performed k-fold cross validation ($k = 5$) for binary classification. Sensitivity (recall), specificity, positive predictive value (PPV, precision), negative predictive value (NPV), and F1 score (a harmonic mean between sensitivity and positive predictive value) were calculated. We performed four experiments with different input variables: (I) using only the summary paragraph parts of discharge letters (conclusion), (II) using the entire corpus of discharge letters, (III) using the entire corpus of discharge letter and adding the variables age and sex, and (IV) multi-label classification of experiment III. For an administrative support tool, it is important to suggest the right diagnoses, ranked by the prediction probabilities. For multi-label assessment we considered every ICD label above a probability threshold as a positive. We assigned this threshold in such a way that the label cardinality for the test set is similar to the label cardinality in the training set.

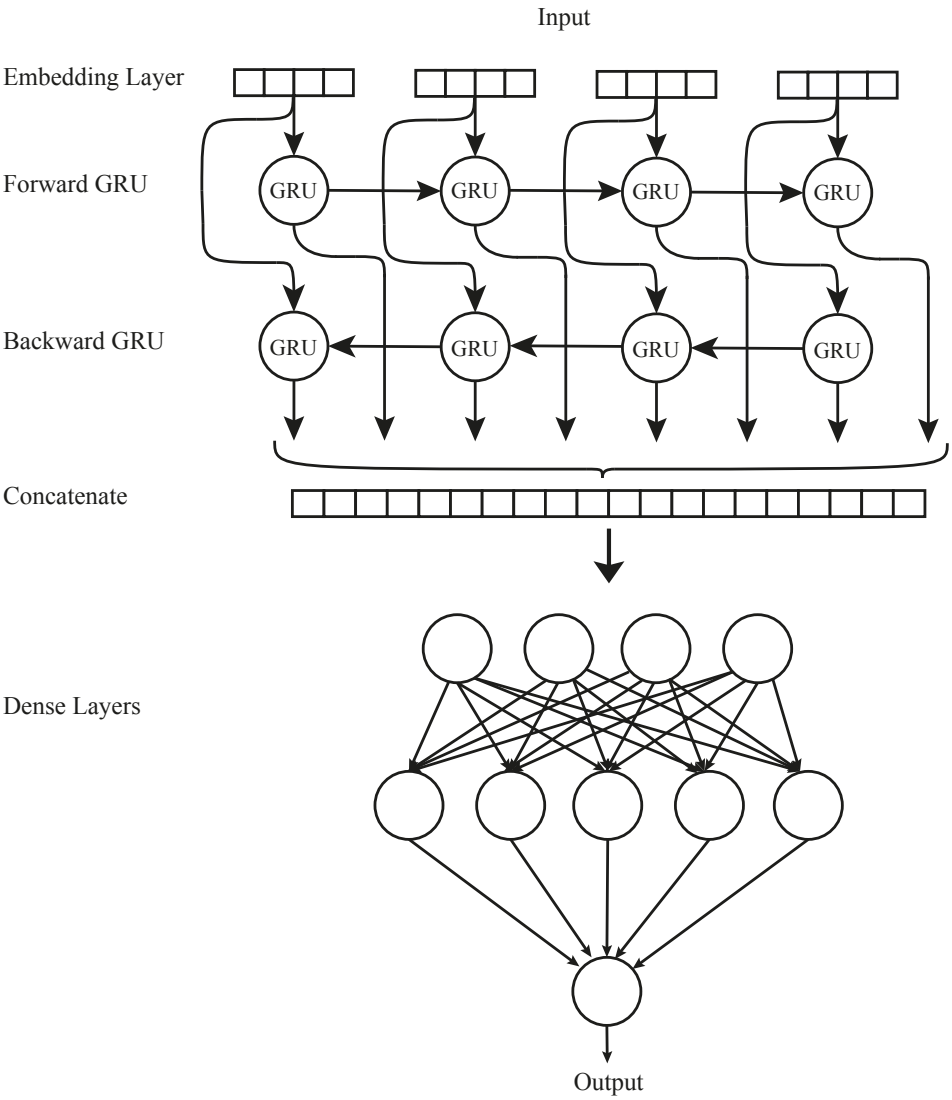


Figure 6.4: Bidirectional gated recurrent unit neural network (BGRU).

6.5 Data Availability

The data set is not publicly available due to patient privacy restrictions.

6.6 Code Availability

The code used in this study can be found at:
`https://github.com/bagheria/cardio-icd-assignment`

Compliance with Ethical Standards

Funding:

Arjan Sammani is supported by the Alexandre Suerman Stipendium and CVON 2015-12 eDETECT YTP. Anneline SJM te Riele is supported by the Dutch Heart Foundation (2015T058), the UMC Utrecht Fellowship Clinical Research Talent and CVON 2015-12 eDETECT. Annette F Baas is funded by Netherlands Heart Foundation (Dekker 2015T041). Folkert W. Asselbergs is supported by UCL Hospitals NIHR Biomedical Research Centre.

Conflict of Interest: The authors declare that they have no conflict of interest.

Acknowledgments: We thank Leslie Beks, Danielle Klokman and Annemiek Tuntelder for their efforts as correctors and medical coders without whom this ICD-10 dataset would not have existed.

Chapter 7

Discussion

7.1 Big Data in Health

Data science has long been recognized as carrying great potential to improve people's lives and gain insight.

Daniel L. Oberski

Over the last decade, there has been a rapid digitalization across the healthcare industries. The application of data science and big data analytics in healthcare has tremendous potential to improve patient daily care, reduce expenditure and reinforce the work of healthcare providers. There is a limited human ability to process this data without automated decision support systems, that is intended to improve healthcare delivery by enhancing medical decisions. This creates the need for integration of data science into the healthcare. Data science has the ability to analyze a wide variety of complex data and generate valuable insights which otherwise would not have been possible. When applied to the healthcare data, it has the potential to identify patterns and lead to improved healthcare quality, reduced costs, and enable accurate decision making. Using Big Data technology, hidden knowledge can be discovered using automated analysis of electronic data in health.

7.2 The Necessity of Text Mining

Classification lies at the heart of both human and machine intelligence.

Daniel Jurafsky, James H. Martin

Healthcare and clinical practice generate huge amounts of text detailing symptoms, test results, diagnoses, conclusions, treatments, and outcomes for patients. This clinical text, documented in health records, is a potential underused source of information for improving healthcare applications. To potentially improve the quality of care, text mining is an effort to make this narrative information accessible for further processing.

The focus of this thesis has been the mining of clinical text in the domain of cardiology, with the aim to develop and evaluate methods for classifying relevant information from such texts. Text mining has previously been applied to a number of tasks within the cardiology domain such as document clustering and topic detection of cardiology reports (Pérez et al., 2018). Text processing has also been used for information extraction from cardiovascular notes (Sammani et al., 2019), and the automated identification and subphenotyping of patients with preserved ejection fraction (Jonnalagadda et al., 2017). An additional area where mining of health records can be applied is in risk prediction models. In this area, text mining and natural language processing (NLP) have been applied for cardiovascular risk factors identification from the clinical notes (Khalifa & Meystre, 2015).

In this thesis, we applied text mining, NLP, and the state-of-the-art machine learning techniques for texts in the cardiology domain. Chapter 2 used a topic modeling approach to detect patients' cardiovascular disease history. Chapter 3 proposed a text mining software package for classifying clinical short notes, tested on a cardiology data set. Chapter 4 applied a deep learning method on chest x-ray reports for the prediction of major cardiovascular events. Chapter 5 studied the ICD-10 coding classification problem for cardiology discharge summaries. And finally, Chapter 6 addressed the problem of multi-label classification for disease coding using the cardiology discharge letters.

7.3 Complexity versus Interpretability

I think you should be more explicit, here in Step 2.

Sidney Harris

Data science has great potential for improving clinical products, processes and research. But a major obstacle to the adoption of data science is that insights about the data and the task the computer solves is hidden in increasingly complex models. Recently in data science applications, more and more attention is focussed on the interpretability of the machine learning models, mainly for the deep learning techniques. Interpretability of the learning models is one of those aspects that is critical in the practical virtue of a data science process and it ensures that the model is aligned with the task. The deep learning models are infamous for their black-box nature due to the vast number of parameters and the complex approach to extracting and combining features. As the deep learning models are able to obtain outstanding performance on clinical text mining tasks, being able to properly interpret the findings is an essential part of the data science pipeline. Model interpretability goes down with the increase in the model complexity; put differently, the easier the model the wider its acceptance. The bag-of-words models are a good example of such easily interpretable models (see Chapter 2, 3, and 4). However, even for the black-box models such as deep learning, techniques exist to improve the interpretability. Feature importance is an example approach for interpreting a deep learning model. For instance, in Chapter 6, we attempted to provide insight into the recurrent neural network model by illustrating the importance of words for the task at hand. Such techniques make the complex deep learning model interpretable and capable of explaining a single prediction or the entire model's behavior.

7.4 Scientific Collaboration

The whole is greater than the sum of its parts.

Aristotle

When individual scientists are connected together to form a team, their achievement trumps many great individuals working separately. Scientific collaboration has the potential to solve complex scientific problems. The last decade, physicians and medical scientists rely more and more on decisions made by expert systems. To this end, they increasingly cooperate with bio-statisticians, data and computer

scientists, software engineers, and IT specialists for the creation of new automated systems to manage the vast amount of data in EHRs. Despite its benefits, interdisciplinary scientific collaboration inevitably involves potential risks and challenges. Some such challenges originate from differences in theoretical and methodological approaches across lines of research. Therefore, scientific collaborations are interpersonal endeavors which require trust, communication, and open-mindedness. These collaborations allow for a complicated and expanded approach to questions within the specific science domains.

7.5 Future Directions

Those who can imagine anything, can create the impossible.

Alan Turing

The final remarks of this thesis concern the further application of data science and text mining techniques in healthcare and some recommendations for future research. There are several interesting avenues to explore in future. I discuss them below.

Clinical Natural Language Processing: As the amount of unstructured text narratives that healthcare systems produce grows, so does the need to intelligently process it and extract different types of knowledge from it. This thesis has demonstrated the importance of clinical narrative text in the healthcare process. However, most of the current advances in text mining and NLP are not applicable in healthcare practice, mainly because of the domain- and language-specific barriers. In the future, with an active role of the health community, clinical NLP should be deployed in practice to accurately recognize the knowledge within clinical text, and feed this knowledge automatically into patient daily care.

Data Sharing: Healthcare information exchange can benefit both healthcare providers and patients. Because of privacy concerns, healthcare organizations have been extremely reluctant to allow access to care data for researchers from outside the associated institutions. Such restricted access to data has hindered collaboration and information exchange among research groups. Recently, by the introduction of technologies such as differential privacy and federated learning (Price & Cohen, 2019; Konečný et al., 2016), we expect the increase in data sharing, facilitating collaboration, and external validity of analysis using integrated data of multiple healthcare organizations. The extent of data sharing required

for widespread adoption of data science and specifically text mining technologies across health systems will require extensive collaborative efforts.

Transfer Learning in Health: In NLP as well as in many areas of machine learning, the standard way to train a model is to annotate a number of examples that are then provided to the model. Recent deep learning-based transfer learning methods have achieved remarkable successes on a wide range of NLP tasks. Given the lack of annotated data sets for training and benchmarking in clinical text mining, in the future, it is expected that the knowledge from related tasks or domains are combined. We also expect, for the NLP tasks in healthcare, more effective approaches that combine semi-supervised learning with transfer learning.

Responsible Text Mining: Artificial intelligence technologies are increasingly prevalent in healthcare applications. The limited adoption of text mining applications in deployment in patient care to date indicates that the current strategies can be improved. Deploying a text mining pipeline within a real world healthcare setting can be substantially more difficult than developing a model in a limited experimental setting. We believe a critical step in the use of clinical decision support systems is to test the system effectively and extensively, before integrating in patient care, in which analyses are exposed to clinical experts but not acted upon.

Creation of Interpretable Systems: Different text mining approaches can complement physicians in the different tasks involved in decision making by clinical narratives. These methods can be computationally very intensive due to the their complexity where traditional approaches does not have the inherent capacity for implementing scalable solutions. But these systems need to be interpretable and supervised by physicians and clinical researchers, who are often more aware of the context.

Team Activity versus a Subspecialty in Healthcare: The clinical systems of today are great advances from what were available a decade ago but can still be improved. Few physicians and healthcare professionals have had training in data science, programming, big data analytics, and other relevant skills to medical informatics. As noted in the previous section, it takes a concerted effort of researchers in healthcare, statistics, data science, machine learning, text mining, and software and IT engineering to bring us towards a future where computers can finally handle all the pragmatic factors in implementing clinical decision support systems.

References

Aggarwal, C. (2018). *Machine learning for text.* Springer.

Alex, B., Grover, C., Tobin, R., Sudlow, C., Mair, G., & Whiteley, W. (2019). Text mining brain imaging reports. *Journal of Biomedical Semantics*, *10*(1), 23–34.

Atutxa, A., de Ilarraza, A., Gojenola, K., Oronoz, M., & Perez de Viñaspre, O. (2019). Interpretable deep learning to map diagnostic texts to ICD-10 codes. *International Journal of Medical Informatics*, *129*, 49–59.

Baghdadi, Y., Bourrée, A., Robert, A., Rey, G., Gallay, A., Zweigenbaum, P., ... Fouillet, A. (2019). Automatic classification of free-text medical causes from death certificates for reactive mortality surveillance in France. *International Journal of Medical Informatics*, *131*, 103915.

Bagheri, A., Sammani, A., Van der Heijden, P. G. M., Asselbergs, F. W., & Oberski, D. L. (2020a). Automatic ICD-10 classification of diseases from Dutch discharge letters. In *13th international joint conference on biomedical engineering systems and technologies* (pp. 281–289).

Bagheri, A., Sammani, A., Van der Heijden, P. G. M., Asselbergs, F. W., & Oberski, D. L. (2020b). ETM: Enrichment by topic modeling for automated clinical sentence classification to detect patients' disease history. *Journal of Intelligent Information Systems*, *55*(2), 329-349.

Bagheri, A., Saraee, M., & De Jong, F. (2014). ADM-LDA: An aspect detection model based on topic modelling using the structure of review sentences. *Journal of Information Science*, *40*(5), 621–636.

Banerjee, I., Ling, Y., Chen, M. C., Hasan, S. A., Langlotz, C. P., Moradzadeh, N., ... others (2019). Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. *Artificial Intelligence in Medicine*, *97*, 79–88.

Baumel, T., Nassour-Kassis, J., Cohen, R., Elhadad, M., & Elhadad, N. (2018). Multi-label classification of patient notes a case study on ICD code assignment. In *AAAI workshops at the thirty-second conference on artificial intelligence.*

Bengio, S., Dembczynski, K., Joachims, T., Kloft, M., & Varma, M. (2019). Extreme classification (dagstuhl seminar 18291). In *Dagstuhl reports* (Vol. 8).

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit.* O'Reilly Media, Inc.

Blanco, A., Casillas, A., Pérez, A., & de Ilarraza, A. (2019). Multi-label clinical document classification: Impact of label-density. *Expert Systems with Applications*, *138*, 112835.

Blanco, A., Perez-de Viñaspre, O., Pérez, A., & Casillas, A. (2020). Boosting ICD multi-label classification of health records with contextual embeddings and label-granularity. *Computer Methods and Programs in Biomedicine*, *188*, 105264.

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*(1), 993-1022.

Bollegala, D., Atanasov, V., Maehara, T., & Kawarabayashi, K. (2018). ClassiNet–Predicting missing features for short-text classification. *arXiv preprint arXiv:1804.05260*.

Boytcheva, S. (2011). Automatic matching of ICD-10 codes to diagnoses in discharge letters. In *Proceedings of the second workshop on biomedical natural language processing* (pp. 11–18).

Branco, P., Torgo, L., & Ribeiro, R. (2015). A survey of predictive modelling under imbalanced distributions. *arXiv preprint arXiv:1505.01658*.

Bui, D., & Zeng-Treitler, Q. (2014). Learning regular expressions for clinical text classification. *Journal of the American Medical Informatics Association*, *21*(5), 850–857.

Buuren, S. v., & Groothuis-Oudshoorn, K. (2010). MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 1–68.

Byrd, R., Steinhubl, S., Sun, J., Ebadollahi, S., & Stewart, W. (2014). Automatic identification of heart failure diagnostic criteria, using text analysis of clinical notes from electronic health records. *International Journal of Medical Informatics*, *83*(12), 983–992.

Cao, L., Gu, D., Ni, Y., & Xie, G. (2019). Automatic ICD code assignment based on ICD's hierarchy structure for chinese electronic medical records. In *AMIA summits on translational science proceedings* (pp. 417–424). American Medical Informatics Association.

Cao, S., Qian, B., Yin, C., Li, X., Wei, J., Zheng, Q., & Davidson, I. (2017). Knowledge guided short-text classification for healthcare applications. In *Proceedings of IEEE international conference on data mining (ICDM)* (pp. 31–40).

Chapman, B. E., Lee, S., Kang, H. P., & Chapman, W. W. (2011). Document-level classification of CT pulmonary angiography reports based on an extension of the ConText algorithm. *Journal of Biomedical Informatics*, *44*(5), 728–737.

Chen, M., Jin, X., & Shen, D. (2011). Short text classification improved by learning multi-granularity topics. In *AAAI, twenty-second international joint conference on artificial intelligence* (pp. 1776–1781).

Chen, M. C., Ball, R. L., Yang, L., Moradzadeh, N., Chapman, B. E., Larson, D. B., ... Lungren, M. P. (2018). Deep learning to classify radiology free-text reports. *Radiology*, *286*(3), 845–852.

Chen, W., Lu, Z., You, L., Zhou, L., Xu, J., & Chen, K. (2020). Artificial intelligence–based multimodal risk assessment model for surgical site infection (AMRAMS): Development and validation study. *JMIR Medical Informatics*, *8*(6), e18186.

Chen, Y., Lu, H., & Li, L. (2017). Automatic ICD-10 coding algorithm using an improved longest common subsequence based on semantic similarity. *PloS One*, *12*(3), e0173410.

Cheng, X., Yan, X., Lan, Y., & Guo, J. (2014). BTM: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, *26*(12), 2928–2941.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Choi, E., Bahadori, M., Schuetz, A., Stewart, W., & Sun, J. (2016). Doctor AI: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference* (pp. 301–318).

Chollet, F., et al. (2015). *Keras.* Retrieved from `https://keras.io`

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cohen, W. (1998). Integration of heterogeneous databases without common domains using queries based on textual similarity. In *ACM SIGMOD record* (Vol. 27, pp. 201–212).

Dai, Z., Sun, A., & Liu, X. (2013). Crest: Cluster-based representation enrichment for short text classification. In *Springer, Pacific-Asia conference on knowledge discovery and data mining* (pp. 256–267).

Demner-Fushman, D., Chapman, W., & McDonald, C. (2009). What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, *42*(5), 760–772.

Drozdov, I., Forbes, D., Szubert, B., Hall, M., Carlin, C., & Lowe, D. (2020). Supervised and unsupervised language modelling in chest x-ray radiological reports. *PLOS One*, *15*(3), e0229963.

Du, J., Chen, Q., Peng, Y., Xiang, Y., Tao, C., & Lu, Z. (2019). ML-Net: Multi-label classification of biomedical texts with deep neural networks. *Journal of the American Medical Informatics Association*, *26*(11), 1279–1285.

Duarte, F., Martins, B., Pinto, C. S., & Silva, M. J. (2018). Deep neural models for ICD-10 coding of death certificates and autopsy reports in free-text. *Journal of Biomedical Informatics*, *80*, 64–77.

Eisenstein, J. (2018). *Natural language processing.* MIT Press.

Feinerer, I. (2019). *Introduction to the tm package: Text mining in R.*

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis.*

Fleuren, W. W., & Alkema, W. (2015). Application of text mining in the biomedical domain. *Methods*, *74*, 97–106.

Fodeh, S., Finch, D., Bouayad, L., Luther, S., Ling, H., Kerns, R., & Brandt, C. (2018). Classifying clinical notes with pain assessment using machine learning. *Medical & Biological Engineering & Computing*, *56*(7), 1285–1292.

Friedman, C., Alderson, P. O., Austin, J. H., Cimino, J. J., & Johnson, S. B. (1994). A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, *1*(2), 161–174.

Friedman, C., Shagina, L., Lussier, Y., & Hripcsak, G. (2004). Automated encoding of clinical documents based on natural language processing. *Journal of the American Medical Informatics Association*, *11*(5), 392–402.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1) (No. 10). Springer Series in Statistics New York.

Gal, Y., & Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems* (pp. 1019–1027).

Gargiulo, F., Silvestri, S., & Ciampi, M. (2018). Deep convolution neural network for extreme multi-label text classification. In *Proceedings of the 11th international joint conference on biomedical engineering systems and technologies (healthinf 2018)* (pp. 641–650).

Ghassemi, M., Naumann, T., Doshi-Velez, F., Brimmer, N., Joshi, R., Rumshisky, A., & Szolovits, P. (2014). Unfolding physiological state: Mortality modelling in intensive care units. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 75–84).

Gong, T., Tan, C. L., Leong, T. Y., Lee, C. K., Pang, B. C., Lim, C. T., ... Zhang, Z. (2008). Text mining in radiology reports. In *2008 eighth IEEE international conference on data mining* (pp. 815–820).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press.

Hearst, M. A. (1999). Untangling text data mining. In *Proceedings of the 37th annual meeting of the association for computational linguistics* (pp. 3–10).

Hill, F., Cho, K., & Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

Hirsch, J., Nicola, G., McGinty, G., Liu, R., Barr, R., Chittle, M., & Manchikanti, L. (2016). ICD-10: history and context. *American Journal of Neuroradiology*, *37*(4), 596–599.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hripcsak, G., Austin, J. H., Alderson, P. O., & Friedman, C. (2002). Use of natural language processing to translate clinical information from a database of 889,921 chest radiographic reports. *Radiology*, *224*(1), 157–163.

Huang, J., Osorio, C., & Sy, L. W. (2019). An empirical evaluation of deep learning for ICD-9 code assignment using MIMIC-III clinical notes. *Computer Methods and Programs in Biomedicine*, *177*, 141–153.

Huang, K., Altosaar, J., & Ranganath, R. (2019). ClinicalBERT: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Hughes, M., Li, I., Kotoulas, S., & Suzumura, T. (2017). Medical text classification using convolutional neural networks. *Studies in Health Technology and Informatics*, *235*, 246–250.

Jackson, R., Kartoglu, I., Stringer, C., Gorrell, G., Roberts, A., Song, X., ... Lewsley, D. (2018). CogStack-experiences of deploying integrated information retrieval and extraction services in a large national health service foundation trust hospital. *BMC Medical Informatics and Decision Making*, *18*(1), 1–13.

Jagannatha, A. N., & Yu, H. (2016). Bidirectional RNN for medical event detection in electronic health records. In *Proceedings of the association for computational linguistics conference* (Vol. 2016, p. 473).

Jamian, L., Wheless, L., Crofford, L. J., & Barnado, A. (2019). Rule-based and machine learning algorithms identify patients with systemic sclerosis accurately in the electronic health record. *Arthritis Research & Therapy*, *21*(1), 305.

Jensen, P. B., Jensen, L. J., & Brunak, S. (2012). Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, *13*(6), 395–405.

Jin, M., Bahadori, M. T., Colak, A., Bhatia, P., Celikkaya, B., Bhakta, R., ... others (2018). Improving hospital mortality prediction with medical named entities and multimodal learning. *arXiv preprint arXiv:1811.12276*.

Jones, O., Maillardet, R., & Robinson, A. (2014). *Introduction to scientific programming and simulation using R*. CRC Press.

Jonnagaddala, J., Liaw, S., Ray, P., Kumar, M., Chang, N., & Dai, H. (2015). Coronary artery disease risk assessment from unstructured electronic health records using text mining. *Journal of Biomedical Informatics*, *58*, S203–S210.

Jonnalagadda, S., Adupa, A., Garg, R., Corona-Cox, J., & Shah, S. (2017). Text mining of the electronic health record: An information extraction approach for automated identification and subphenotyping of HFPEF patients for clinical trials. *Journal of Cardiovascular Translational Research*, *10*(3), 313–321.

Jurafsky, D., & Martin, J. (2019). *Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing* (3rd ed.). Prentice Hall.

Kalyan, K. S., & Sangeetha, S. (2020). SECNLP: A survey of embeddings in clinical natural language processing. *Journal of Biomedical Informatics*, *101*, 103323.

Karimi, S., Dai, X., Hassanzadeh, H., & Nguyen, A. (2017). Automatic diagnosis coding of radiology reports: A comparison of deep learning and conventional classification methods. In *Proceedings of the BioNLP 2017 workshop, association for computational linguistics* (pp. 328–332).

Kemp, J., Rajkomar, A., & Dai, A. M. (2019). Improved hierarchical patient classification with language model pretraining over clinical notes. *arXiv preprint arXiv:1909.03039*.

Khalifa, A., & Meystre, S. (2015). Adapting existing natural language processing resources for cardiovascular risk factors identification in clinical notes. *Journal of Biomedical Informatics*, *58*, S128–S132.

Khoo, A., Marom, Y., & Albrecht, D. (2006). Experiments with sentence classification. In *Proceedings of the Australasian language technology workshop* (pp. 18–25).

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kocbek, S., Cavedon, L., Martinez, D., Bain, C., Mac Manus, C., Haffari, G., ... Verspoor, K. (2016). Text mining electronic hospital records to automatically classify admissions against disease: measuring the impact of linking data sources. *Journal of Biomedical Informatics*, *64*, 158–167.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. In *NIPS workshop*.

Koopman, B., Karimi, S., Nguyen, A., McGuire, R., Muscatello, D., Kemp, M., ... Thackway, S. (2015). Automatic classification of diseases from free-text death certificates for real-time surveillance. *BMC Medical Informatics and Decision Making*, *15*(1), 53.

Koopman, B., Zuccon, G., Nguyen, A., Bergheim, A., & Grayson, N. (2015). Automatic ICD-10 classification of cancers from free-text death certificates. *International Journal of Medical Informatics*, *84*(11), 956–965.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150.

Kozlowski, M., & Rybinski, H. (2017). Semantic enriched short text clustering. In *International symposium on methodologies for intelligent systems* (pp. 435–445).

Kozlowski, M., & Rybinski, H. (2019). Clustering of semantically enriched short texts. *Journal of Intelligent Information Systems*, *53*(1), 69–92.

Kraaij, W., & Pohlmann, R. (1994). Porter's stemming algorithm for Dutch. *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie*, 167–180.

Laserson, J., Lantsman, C. D., Cohen-Sfady, M., Tamir, I., Goz, E., Brestel, C., ... Elnekave, E. (2018). TextRay: Mining clinical reports to gain a broad

understanding of chest x-rays. In *International conference on medical image computing and computer-assisted intervention* (pp. 553–561).

Lee, J., & Dernoncourt, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*.

Lin, Y.-W., Zhou, Y., Faghri, F., Shaw, M. J., & Campbell, R. H. (2019). Analysis and prediction of unplanned intensive care unit readmission using recurrent neural networks with long short-term memory. *PLOS One*, *14*(7), e0218942.

Liu, J., Zhang, Z., & Razavian, N. (2018). Deep EHR: Chronic disease prediction using medical notes. *arXiv preprint arXiv:1808.04928*.

Lv, Y., Deng, Y., Liu, M., Cui, Y., & Lu, Q. (2016). Short text classification of EMR based on entities and dependency parser. *Chinese Journal of Medical Instrumentation*, *40*(4), 245–249.

Mehta, N., & Pandit, A. (2018). Concurrence of big data analytics and healthcare: A systematic review. *International Journal of Medical Informatics*, *114*, 57–65.

Menger, V., Scheepers, F., van Wijk, L., & Spruit, M. (2018). DEDUCE: A pattern matching method for automatic de-identification of Dutch medical text. *Telematics and Informatics*, *35*(4), 727–736.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Karafiát, M., Burget, L., Cernocky, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association* (pp. 1045–1048).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Miranda, R., Martins, B., Silva, M., Silva, N., & Leite, F. (2018). Deep learning for multi-label ICD-9 classification of hospital discharge summaries. *Thesis report, University of Lisbon, Portugal*.

Mirończuk, M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, *106*, 36–54.

Monshi, M. M. A., Poon, J., & Chung, V. (2020). Deep learning in generating radiology reports: A survey. *Artificial Intelligence in Medicine*, 101878.

Moradi, M., Dorffner, G., & Samwald, M. (2020). Deep contextualized embeddings for quantifying the informative content in biomedical text summarization. *Computer Methods and Programs in Biomedicine*, *184*, 105117.

Mujtaba, G., Shuib, L., Idris, N., Hoo, W., Raj, R., Khowaja, K., ... Nweke, H. (2019). Clinical text classification research trends: Systematic literature review and open issues. *Expert Systems with Applications*, *116*, 494–520.

Mujtaba, G., Shuib, L., Raj, R. G., Rajandram, R., Shaikh, K., & Al-Garadi, M. A. (2017). Automatic ICD-10 multi-class classification of cause of death from plaintext autopsy reports through expert-driven feature selection. *PloS One*, *12*(2), e0170242.

Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., & Eisenstein, J. (2018). Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT press.

Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulos, A., ... others (2017). Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Nguyen, A. N., Truran, D., Kemp, M., Koopman, B., Conlan, D., O'Dwyer, J., ... others (2018). Computer-assisted diagnostic coding: effectiveness of an NLP-based approach using SNOMED CT to ICD-10 mappings. In *AMIA annual symposium proceedings* (p. 807).

Nigam, P. (2016). *Applying deep learning to ICD-9 multi-label classification from medical records* (Tech. Rep.). Technical Report, Stanford University.

Pakhomov, S. V., Buntrock, J. D., & Chute, C. G. (2006). Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. *Journal of the American Medical Informatics Association*, *13*(5), 516–525.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pérez, J., Pérez, A., Casillas, A., & Gojenola, K. (2018). Cardiology record multi-label classification using latent Dirichlet allocation. *Computer Methods and Programs in Biomedicine*, *164*, 111–119.

Perotte, A., Pivovarov, R., Natarajan, K., Weiskopf, N., Wood, F., & Elhadad, N. (2014). Diagnosis code assignment: Models and evaluation metrics. *Journal of the American Medical Informatics Association*, *21*(2), 231–237.

Pons, E., Braun, L. M., Hunink, M. M., & Kors, J. A. (2016). Natural language processing in radiology: A systematic review. *Radiology*, *279*(2), 329–343.

Porter, M. F. (2001). *Snowball: A language for stemming algorithms.*

Price, W. N., & Cohen, I. G. (2019). Privacy in the age of medical big data. *Nature Medicine*, *25*(1), 37–43.

Reed, C. (2012). Latent Dirichlet allocation: Towards a deeper understanding. *Available at obphio.us*, 1–13.

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks* (pp. 45–50).

Ruder, S. (2019). Neural transfer learning for natural language processing. *PhD Thesis, NUI Galway*.

Sammani, A., Jansen, M., Linschoten, M., Bagheri, A., de Jonge, N., Kirkels, H., ... others (2019). UNRAVEL: Big data analytics research data platform to improve care of patients with cardiomyopathies using routine electronic health records and standardised biobanking. *Netherlands Heart Journal*, *27*(9), 426–434.

Savova, G., Masanz, J., Ogren, P., Zheng, J., Sohn, S., Kipper-Schuler, K., & Chute, C. (2010). Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, *17*(5), 507–513.

Scheurwegs, E., Luyckx, K., Luyten, L., Daelemans, W., & Van den Bulcke, T. (2016). Data integration of structured and unstructured sources for assigning clinical codes to patient stays. *Journal of the American Medical Informatics Association*, *23*(e1), e11–e19.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

Sevenster, M., Bozeman, J., Cowhy, A., & Trost, W. (2015). A natural language processing pipeline for pairing measurements uniquely across free-text CT reports. *Journal of Biomedical Informatics*, *53*, 36–48.

Sheikhalishahi, S., Miotto, R., Dudley, J. T., Lavelli, A., Rinaldi, F., & Osmani, V. (2019). Natural language processing of clinical notes on chronic diseases: Systematic review. *JMIR Medical Informatics*, *7*(2), e12239.

Shen, Y., Zhang, Q., Zhang, J., Huang, J., Lu, Y., & Lei, K. (2018). Improving medical short text classification with semantic expansion using word-cluster embedding. In *International conference on information science and applications* (pp. 401–411).

Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2017). Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE Journal of Biomedical and Health Informatics*, *22*(5), 1589–1604.

Shin, B., Chokshi, F. H., Lee, T., & Choi, J. D. (2017). Classification of radiology reports using neural attention models. In *2017 international joint conference on neural networks (ijcnn)* (pp. 4363–4370).

Shing, H.-C., Wang, G., & Resnik, P. (2019). Assigning medical codes at the encounter level by paying attention to documents. *arXiv preprint arXiv:1911.06848*.

Silge, J., & Robinson, D. (2016). tidytext: Text mining and analysis using tidy data principles in R. *Journal of Open Source Software*, *1*(3), 37.

Simons, P. C. G., Algra, A., Van de Laak, M., Grobbee, D., & Van der Graaf, Y. (1999). Second manifestations of arterial disease (SMART) study: Rationale and design. *European Journal of Epidemiology*, *15*(9), 773–781.

Smit, A., Jain, S., Rajpurkar, P., Pareek, A., Ng, A. Y., & Lungren, M. P. (2020). CheXbert: Combining automatic labelers and expert annotations for accurate radiology report labeling using BERT. *arXiv preprint arXiv:2004.09167*.

Sohn, S., Clark, C., Halgrim, S., Murphy, S., Chute, C., & Liu, H. (2014). MedXN: An open source medication extraction and normalization tool for clinical text. *Journal of the American Medical Informatics Association*, *21*(5), 858–865.

Soysal, E., Wang, J., Jiang, M., Wu, Y., Pakhomov, S., Liu, H., & Xu, H. (2017). CLAMP–A toolkit for efficiently building customized clinical natural language processing pipelines. *Journal of the American Medical Informatics Association*, *25*(3), 331–336.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., & Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval* (pp. 841–842).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Stausberg, J., Lehmann, N., Kaczmarek, D., & Stein, M. (2008). Reliability of diagnoses coding with ICD-10. *International Journal of Medical Informatics*, *77*(1), 50–57.

Suresh, H., Hunt, N., Johnson, A., Celi, L. A., Szolovits, P., & Ghassemi, M. (2017). Clinical intervention prediction and understanding using deep networks. *arXiv preprint arXiv:1705.08498*.

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning (icml)* (pp. 1017–1024).

Sutton, R. T., Pincock, D., Baumgart, D. C., Sadowski, D. C., Fedorak, R. N., & Kroeker, K. I. (2020). An overview of clinical decision support systems: Benefits, risks, and strategies for success. *NPJ Digital Medicine*, *3*(1), 1–10.

Taira, R. K., Soderland, S. G., & Jakobovits, R. M. (2001). Automatic structuring of radiology free-text reports. *Radiographics*, *21*(1), 237–245.

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine learning Research*, *2*(Nov), 45–66.

Torii, M., Fan, J., Yang, W., Lee, T., Wiley, M., Zisook, D., & Huang, Y. (2015). Risk factor detection for heart disease by applying text analytics in electronic medical records. *Journal of Biomedical Informatics*, *58*, S164–S170.

Unnikrishnan, P., Govindan, V., & Kumar, S. M. (2019). Enhanced sparse representation classifier for text classification. *Expert Systems with Applications*, *129*, 260–272.

van Amsterdam, W., Verhoeff, J., de Jong, P., Leiner, T., & Eijkemans, M. (2019). Eliminating biasing signals in lung cancer images for prognosis predictions with deep learning. *NPJ Digital Medicine*, *2*(1), 1–6.

van Kesteren, E.-J. (2020, Jan). Vankesteren/firatheme: Firatheme version 0.2.1. doi: 10.5281/zenodo.3604681

Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., ... others (2018). Clinical information extraction applications: A literature review. *Journal of Biomedical Informatics*, *77*, 34–49.

Wang, Z., Shah, A. D., Tate, A. R., Denaxas, S., Shawe-Taylor, J., & Hemingway, H. (2012). Extracting diagnoses and investigation results from unstructured text in electronic health records by semi-supervised machine learning. *PLOS One*, *7*(1), e30412.

Weeks, H., Beck, C., McNeer, E., Bejan, C., Denny, J., & Choi, L. (2019). medExtractR: A medication extraction algorithm for electronic health records using the R programming language. *medRxiv*, 19007286.

Wood, D., Lynch, J., Kafiabadi, S., Guilhem, E., Busaidi, A., Montvila, A., ... others (2020). Automated labelling using an attention model for radiology reports of MRI scans (ALARM). *arXiv preprint arXiv:2002.06588*.

Wu, H., Toti, G., Morley, K., Ibrahim, Z., Folarin, A., Jackson, R., ... others (2018). SemEHR: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research. *Journal of the American Medical Informatics Association*, *25*(5), 530–537.

Wu, X., Zhao, Y., Radev, D., & Malhotra, A. (2020). Identification of patients with carotid stenosis using natural language processing. *European Radiology*, 1–9.

Xiao, C., Choi, E., & Sun, J. (2018). Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, *25*(10), 1419–1428.

Xie, X., Xiong, Y., Yu, P. S., & Zhu, Y. (2019). EHR coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 649–658).

Xu, K., Lam, M., Pang, J., Gao, X., Band, C., Xie, P., & Xing, E. (2018). Multimodal machine learning for automated ICD coding. *arXiv preprint arXiv:1810.13348*.

Yang, S., Huang, G., & Cai, B. (2019). Discovering topic representative terms for short text clustering. *IEEE Access*, *7*, 92037–92047.

Yang, S., Lu, W., Yang, D., Yao, L., & Wei, B. (2015). Short text understanding by leveraging knowledge into topic model. In *Association for computational linguistics, proceedings of the 2015 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 1232–1237). Denver, Colorado.

Yao, L., Mao, C., & Luo, Y. (2019). Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. *BMC Medical Informatics and Decision Making*, *19*(3), 71.

Yim, W.-w., Yetisgen, M., Harris, W. P., & Kwan, S. W. (2016). Natural language processing in oncology: A review. *JAMA oncology*, *2*(6), 797–804.

Yin, C., Shi, L., & Wang, J. (2017). Short text classification technology based on KNN + hierarchy SVM. In *Springer, advanced multimedia and ubiquitous engineering* (pp. 633–639).

Yu, H., Ho, C., Juan, Y., & Lin, C. (2013). Libshorttext: A library for short-text classification and analysis. *Rapport interne, Department of Computer Science, National Taiwan University. Software available at http://www. csie. ntu. edu. tw/cjlin/libshorttext*.

Zech, J., Pain, M., Titano, J., Badgeley, M., Schefflein, J., Su, A., ... Oermann, E. (2018). Natural language–based machine learning models for the annotation of clinical radiology reports. *Radiology*, *287*(2), 570–580.

Zelikovitz, S., & Hirsh, H. (2000). Improving short text classification using unlabeled background knowledge to assess document similarity. In *Proceedings of the seventeenth international conference on machine learning* (pp. 1183–1190).

Zhang, Y., Nie, A., Zehnder, A., Page, R. L., & Zou, J. (2019). VetTag: Improving automated veterinary diagnosis coding via large-scale language modeling. *NPJ Digital Medicine*, *2*(1), 1–8.

Zweigenbaum, P., & Lavergne, T. (2016). Hybrid methods for ICD-10 coding of death certificates. In *Proceedings of the seventh international workshop on health text mining and information analysis* (pp. 96–105).

Nederlandse Samenvatting

Elektronische gezondheidsdossiers (EHR's) zijn rijk aan gegevens met het potentieel om zorg veiliger te maken, medische fouten te verminderen, de uitgaven voor gezondheidszorg te verlagen en om zorgaanbieders in staat te stellen hun productiviteit en efficiëntie te verbeteren. Een groot deel van deze gegevens is vrije tekst in de vorm van aantekeningen van artsen, ontslagsamenvattingen, radiologierapporten, enzovoorts. Deze klinische tekst volgt de patiënt door de zorgprocedures en documenteert de klachten en symptomen van de patiënt, het lichamelijk onderzoek, de diagnostische tests, de conclusies, de behandelingen en de resultaten van de behandeling.

Ondanks vele pogingen om het gebruik van vrije tekst te verminderen door EHR's beter te structureren wordt er nog steeds veel gebruik gemaakt van vrije tekst door zorgaanbieders. Als alternatief voor het handmatig structureren van de informatie in deze vrije tekst kunnen "tekst mining" technieken worden toegepast om een meer gestructureerde weergave te creëren, waardoor de inhoud ervan toegankelijker wordt voor data science, machine learning en statistiek. Deze thesis is gericht op het bieden van oplossingen voor een aantal van de uitdagingen in de analyse van het veelvoorkomende vrijetekstelement in het klinische domein.

Tekst mining wordt toegepast op tal van medische systemen voor de-identificatie, klinische beslissingsondersteuning, identificatiesystemen voor patiënten, classificatiesystemen voor ziekten, en klinische voorspellingsmodellen. Veel van deze klinische toepassingen zijn afhankelijk van een of andere vorm van *tekstclassificatie*. Tekstclassificatie is een taak die bestaat uit het automatisch toewijzen van een document aan een vooropgestelde set van klassen of labels. De focus van het werk dat in dit proefschrift wordt beschreven is de analyse en classificatie van de klinische vrije tekst die in het Universitair Medisch Centrum (UMC) te Utrecht wordt gebruikt.

In **hoofdstuk 2** wordt de methode voor het eerste onderzoeksprobleem beschreven: de extractie van medische geschiedenis door middel van classificatie van klinische zinnen in vrije text. Vanwege het beperkte aantal woorden dat in klinische zinnen wordt gebruikt, wordt dit probleem beschouwd als een probleem van korte tekstclassificatie. We stellen een *unsupervised topic modelling-based smoothing* methode voor die gebruik maakt van een intern mechanisme voor kennisverwerving zonder gebruik te maken van een extern woordenboek. Deze methode gebruiken we om de weergave van de zinnen te verrijken in een dataset van

klinische cardiovasculaire aantekeningen van de afdeling Cardiologie van het UMC Utrecht.

De aanzienlijke hoeveelheid ongestructureerde korte tekst in gezondheidstoepassingen, met name in klinische cardiovasculaire notities, heeft een dringende behoefte gecreëerd aan instrumenten die specifieke informatie uit tekstrapporten kunnen ontleden. In **hoofdstuk 3** wordt een op Python gebaseerd software pakket (SALTClass) gepresenteerd voor de classificatie van korte en lange klinische teksten. Het SALTClass-pakket is een op machine learning gebaseerde Natural Language Processing (NLP) toolkit. Het bevat verschillende functies voor het voorbewerken, opschonen, clusteren en classificeren van teksten. Het SALTClass-pakket implementeert smoothing methoden op basis van zeven clusteringalgoritmen: LDA, K-Means, MiniBatchK-Means, BIRCH, MeanShift, DBScan en GMM. Smoothing methoden worden toegepast op de resulterende clusterinformatie om de tekstrepresentatie te verrijken. Voor de classificatiestap met de verrijkte tekst zijn ook tien verschillende algoritmes in SALTClass geïntegreerd. Het SALTClass softwarepakket stelt gebruikers in staat om verschillende configuratiecombinaties toe te passen op hun studie. Om de effectiviteit van SALTClass te evalueren, analyseren we cardiovasculaire aantekeningen die in het UMC Utrecht zijn verzameld.

In **hoofdstuk 4** wordt een multimodale architectuur met tekst mining geïntroduceerd om grote cardiovasculaire incidenten te voorspellen. Het doel van dit hoofdstuk is om de waarde van klinische tekstclassificatie aan te tonen wanneer er naast de klassieke klinische gegevens van patiënten ook tekstgegevens beschikbaar zijn. We stellen een deep learning architectuur voor die neurale tekstrepresentatie integreert met voorbewerkte klinische voorspellers voor cardiovasculaire risicovoorspelling. We evalueren de toegevoegde waarde van tekstgegevens uit radiologierapporten voor patiënten met vasculaire aandoeningen of een vasculaire risicofactor, met behulp van de voorgestelde text mining pipeline.

In **hoofdstuk 5** worden state-of-the-art deep learning classificatiesystemen beoordeeld voor internationale classificatie van ziekten en gerelateerde gezondheidsproblemen (ICD-10). Deze deep learning systemen worden vergeleken met baselinesystemen op een dataset die is opgebouwd uit Nederlandse ontslagbrieven voor cardiologie in het UMC Utrecht. De ICD-coderingstaak is uitdagend vanwege het gebruik van vrije tekst, het feit dat meerdere diagnostische codes tegelijk van toepassing kunnen zijn, en het grote aantal mogelijke codes. De methoden die we onderzoeken in dit hoofdstuk zijn support vector machines, convolutional neural networks (CNN), long short term memory (LSTM), bidirectionele LSTM (biLSTM), en een hiërarchische attention-based gated recurrent unit (GRU).

Hoofdstuk 6 heeft als doel de onderzoekslijn in hoofdstuk 5 voort te zetten om een hoogperformante, deep learning pipeline te creëren voor de geautomatiseerde multi-label classificatie van betrouwbare ICD-10-codes in de klinische vrije tekst in het domein van de cardiologie. In dit hoofdstuk richten we ons op de veelgebruikte en goed gedefinieerde drie-cijferige ICD-10-codes. We onderzoeken het gebruik

van alleen de samenvattende paragraaf van ontslagbrieven (conclusie), met het toevoegen van klinische variabelen (leeftijd / geslacht) en multi-label classificatie zoals dat in de klinische praktijk het geval is.

Tenslotte bevat **Hoofdstuk 7** de discussiepunten waarin ik enkele belangrijke overwegingen voor tekst mining in de gezondheidszorg samenvat. In dit hoofdstuk reflecteer ik over het werk in de overige hoofdstukken van dit proefschrift.

Curriculum Vitae

Ferdowsi University of Mashhad 2004 – 2007
BSc in Software Engineering

Isfahan University of Technology 2007 – 2009
MSc in Artificial Intelligence

Isfahan University of Technology 2010 – 2014
PhD in Computer Engineering

University of Twente 2012
Researcher

University of Kashan 2015 – 2017
Researcher

Utrecht University & UMC Utrecht 2017 – 2020
PhD in Methodology and Statistics

Utrecht University 2020 –
Assistant Professor in Applied Data Science

List of Publications

Bagheri, A., Sammani, A., Van der Heijden, P. G. M., Asselbergs, F. W., & Oberski, D. L. (2020). ETM: Enrichment by topic modeling for automated clinical sentence classification to detect patients' disease history. *Journal of Intelligent Information Systems*, 55(2), 329-349.

Bagheri, A., Groenhof, T. K. J., Veldhuis, W. B., De Jong, P. A., Asselbergs, F. W., & Oberski, D. L. (2020). Multimodal learning for cardiovascular risk prediction using EHR data. *In Proceedings of the 11th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB 2020).*

Bagheri, A., Sammani, A., Van der Heijden, P. G. M., Asselbergs, F. W., & Oberski, D. L. (2020). Automatic ICD-10 classification of diseases from Dutch discharge letters. *In Proceeding of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies*, 281-289.

Boeschoten, L., Van Kesteren, E. J., **Bagheri, A.**, & Oberski, D. L. (2020). Fair inference on error-prone outcomes. *ECAI conference 2020 workshop: Artificial Intelligence for a Fair, Just and Equitable World.*

Van de Leur, R. R., Boonstra, M. J., **Bagheri, A.**, Roudijk, R. W., Sammani, A., Taha, K., Doevendans, P. A. F. M., Van der Harst, P., Van Dam, P. M., Hassink, R. J., Van Es, R., & Asselbergs F. W. (2020). Big data and artificial intelligence: Opportunities and threats in electrophysiology. *Arrhythmia & Electrophysiology Review*, 9(3), 146-154.

Bagheri, A., Groenhof, T. K. J., Asselbergs, F. W., Haitjema, S., Bots, M. L., Veldhuis, W. B., De Jong, P. A., & Oberski, D. L. (2020, submitted). Using chest x-ray reports for prediction of recurrence of major cardiovascular events in cardiovascular patients.

Sammani, A., **Bagheri, A.**, Van der Heijden, P. G. M., Te Riele, A., Baas, A., Oberski, D. L., & Asselbergs, F. W. (2020, submitted). Automatic multilabel detection of ICD10 codes in discharge letters from cardiology.

Ferdinands, G., Schram, R., De Bruin, J., **Bagheri, A.**, Oberski, D., Tummers, L., & Van de Schoot, R. (2020, submitted). Interactive screening prioritization in systematic reviews by employing active learning.

Felix, S. E. A., **Bagheri, A.**, Ramjankhan, F. R., Spruit, M. R., Oberski, D. L., De Jonge, N., Van Laake, L. W., Suyker, W. J. L., & Asselbergs, F. W. (2020, submitted). A data mining-based cross-industry process for predicting major bleeding in mechanical circulatory support.

Sammani, A., Jansen, M., Linschoten, M., **Bagheri, A.**, De Jonge, N., Kirkels, H., Van Laake, L., Vink, A., Van Tintelen, J. P., Dooijes, D., Te Riele, A., Harakalova, M., Baas, A. F., & Asselbergs, F. W. (2019). UNRAVEL: Big data analytics research data platform to improve care of patients with cardiomyopathies using routine electronic health records and standardized biobanking. *Netherlands Heart Journal*, 27(9), 426-434.

Bagheri, A., Oberski, D. L., Sammani, A., Van der Heijden, P. G. M., Asselbergs, F. W. (2019). SALTClass: Classifying clinical short notes using background knowledge from unlabeled data. *bioRxiv 801944*.

Bagheri, A. (2019). Integrating word status for joint detection of sentiment and aspect in reviews. *Journal of Information Science*, 45(6), 736-755.

Sedighi, Z., Ebrahimpour-Komleh, H., **Bagheri, A.**, & Kosseim, L. (2019). Learning to identify fake opinion reviews using a neural network with attention. *In Proceeding of the 32nd International Artificial Intelligence Research Society Conference*, 245-248.

Asgarnezhad, R., Monadjemi, S. A., Soltanaghaei, M., & **Bagheri, A.** (2018). SFT: A model for sentiment classification using supervised methods on Twitter. *Journal of Theoretical & Applied Information Technology*, 96(8), 2242-2251.

Afsharizadeh, M., Ebrahimpour-Komleh, H., & **Bagheri, A.** (2018). Query-oriented text summarization using sentence extraction technique. *In Proceeding of the 4th IEEE International Conference on Web Research*, 128-132.

Sedighi, Z., Ebrahimpour-Komleh, H., & **Bagheri, A.** (2017). RLOSD: representation learning based opinion spam detection. *In Proceeding of the 3rd IEEE Conference on Intelligent Systems and Signal Processing*, 74-80.

Zaghian, A., & **Bagheri, A.** (2016). A combined model of clustering and classification methods for preserving privacy in social networks against inference

and neighborhood attacks. *International Journal of Security and Its Applications*, 10(1), 95-102.

Bagheri, A., Saraee, M., & Nadi, S. (2015). PSA: A hybrid feature selection approach for Persian text classification. *Journal of Computing and Security, Special Issue in Data Mining and Knowledge Discovery*, 1(4), 261-272.

Bagheri, A., Saraee, M., & De Jong, F. (2014). ADM-LDA: An aspect detection model based on topic modeling by using structures of reviews. *Journal of Information Science*, 40(5), 621-636.

Bagheri, A., Saraee, M., & De Jong, F. (2014). Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews. *Knowledge-Based Systems*, 52, 201–213.

Bagheri, A., & Saraee, M. (2014). Persian sentiment analyzer: A framework based on a novel feature selection method. *International Journal of Artificial Intelligence*, 12(2), 115-129.

Bagheri, A., Saraee, M., & De Jong, F. (2013). An unsupervised aspect detection model for sentiment analysis of reviews. *Natural Language Processing and Information Systems NLDB2013*, 140-151.

Saraee, M., & **Bagheri, A.** (2013). Feature selection methods in Persian sentiment analysis. *Natural Language Processing and Information Systems NLDB2013*, 303-308.

Bagheri, A., Saraee, M., & De Jong, F. (2013). Latent Dirichlet Markov allocation for sentiment analysis. *In Proceeding of the Fifth European Conference on Intelligent Management Systems in Operations*, 90-96.

Bagheri, A., Saraee, M., & De Jong, F. (2013). Sentiment classification in Persian introducing a mutual information-based method for feature selection. *In Proceedings of the 21st Iranian Conference on Electrical Engineering*, 1-6.

Nadi, S., Saraee, M., & **Bagheri, A.** (2011). A hybrid recommender system for dynamic web users. *International Journal of Multimedia and Image Processing*, 1(1), 3-8.

Moghimi, M., Saraee, M., & **Bagheri, A.** (2011). Modeling batch annealing process using data mining techniques for cold rolled steel sheets. *In Proceedings of the First International Workshop on Data Mining for Service and Maintenance*, 18-22.

Nadi, S., Saraee, M., & **Bagheri, A.** (2010). FARS: Fuzzy ant based recommender system for web users. *International Journal of Computer Science Issues*, 7(6), 203-209.

Norouzzadeh, M., **Bagheri, A.**, & Saraee, M. (2009). Web search personalization: A fuzzy adaptive approach. *In Proceeding of the 2nd IEEE International Conference on Computer Science and Information Technology*, 143-148.

Bagheri, A., Akbarzadeh, M., & Saraee, M. (2008). Finding shortest path with learning algorithms. *International Journal of Artificial Intelligence*, 1(8), 86-95.

Acknowledgments

This thesis would not have been possible without the support of many people. I would like to take this opportunity to thank them.

Above all, I would like to thank my supervisors. Daniel, I have learned over and over from your unique perspective on research, and your profound and fascinating insight on almost any issue. I am very grateful for all your patience, constant support, excellent advice and encouragement. Folkert, I am grateful for your inspirational advice and constant encouragement over the last several years. Peter, ik ben er zeker van dat ik niet zou zijn waar ik nu ben zonder jouw steun, onderwijs en vriendelijkheid. Van jou heb ik ook geleerd dat het belangrijkste om succesvol te zijn in een nieuwe gemeenschap is om de makkelijkste manier van communiceren met hen te leren. Ik ben je daar dankbaar voor. I could not have wished for better supervisors.

I would like to thank the members of the reading committee and the defence committee for their time and effort spent on evaluating this thesis.

I am also grateful to all of my co-authors for the collaborations and the helpful feedback over the years. Collaborating with them has made me a better researcher.

Erik-Jan and Oisín, I am very lucky to have you as my best friends and thanks to I-don't-know-who (most likely Kevin) for putting all three of us together in the magic office, C1.22. Thanks for being there for me, joining me to grab a coffee (though sometimes I forgot to drink mine), and dealing with my silly questions. I am also honored to have you two as my paranymphs in my defence. Many thanks!

I would like to thank the members of the Human Data Science group, Daniel, Erik-Jan, Laura, Anastasia and Qixiang, for the occasional talks that we had during Corona times and for brainstorming ideas.

Thanks to all the colleagues at the Methodology and Statistics department for the welcoming and pleasant environment with friendly staff. Many thanks to Anne, Duco, Erik-Jan, Fayette, Hidde, Jeroen, Karlijn, Kees, Lientje, Oisín, Sanne, Sebastian, and all other colleagues for all the funny and some weird discussions during lunch. Thanks to Gerko for his support and advice over the years. Thanks to Flip, Chantal, Irma, and Els for their continued support. My special thanks to Kevin! In my opinion he is the heart of the department. Believe me because I worked in the department of Heart and Lungs in UMCU and I know what a heart does! Thank you Kevin! I am very lucky that I get to stay in this unbelievably fantastic department.

Thanks to all my colleagues at the hospital, especially Arjan, Jantine, and Katrien. Arjan, I remember the very first conversation that I had with you was in Folkert's office on his phone! From that day you and I started our collaboration, which continued throughout my thesis, and which will continue into the future. Without your support this thesis would not have been possible.

It may sound funny, but I would like to also thank all the cast and crew of the series Friends! Friends is a realistic depiction of life and a good way of oblivion when a paper gets rejected! As Phoebe said: "They don't know that we know they know we know."

I would like to thank my friends and family for being there for me in my life. I wouldn't be where I am today without the support, encouragement and love from my parents.

Finally, thanks to my small family: Shiva and Ronika. Shiva, you are the best thing that ever happened to me. You are one of the sweetest and most kind-hearted people and the reason that I could follow all my dreams. Thank you for always being there for me, and all your patience with me, cheering me up and making me feel the luckiest man in the world. Undoubtedly one of the best days that you and I had was the day Ronika was born. Although she is now very young, we are happy that she is enthusiastic about learning the alphabet and reading books, and it is funny that she is already correcting us in speaking Dutch. Everyday with you two is an exciting adventure for me!